Rcpp Version 0.12.7

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

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 pythag (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 double R::bessel_i (double *x*, double *al*, double *ex*) [inline]

Definition at line 209 of file Rmath.h.

5.1.1.2 double R::bessel_i_ex (double *x*, double *al*, double *ex*, double * *bi*) [inline]

Definition at line 213 of file Rmath.h.

5.1.1.3 double R::bessel_j (double *x*, double *al*) [inline]

Definition at line 210 of file Rmath.h.

5.1.1.4 double R::bessel_j_ex (double *x*, double *al*, double * *bj*) [inline]

Definition at line 214 of file Rmath.h.

5.1.1.5 double R::bessel_k (double *x*, double *al*, double *ex*) [inline]

Definition at line 211 of file Rmath.h.

5.1.1.6 double R::bessel_k_ex (double *x*, double *al*, double *ex*, double * *bk*) [inline]

Definition at line 215 of file Rmath.h.

5.1.1.7 double R::bessel_y (double *x*, double *al*) [inline]

Definition at line 212 of file Rmath.h.

5.1.1.8 double R::bessel_y_ex (double *x,* double *al,* double * *by*) [inline]

Definition at line 216 of file Rmath.h.

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

Definition at line 202 of file Rmath.h.

Referenced by Rcpp::internal::lfactorial().

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

Definition at line 205 of file Rmath.h.

Referenced by Rcpp::internal::lfactorial().

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

Definition at line 60 of file Rmath.h.

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

Definition at line 96 of file Rmath.h.

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

Definition at line 105 of file Rmath.h.

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

Definition at line 72 of file Rmath.h.

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

Definition at line 111 of file Rmath.h.

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

Definition at line 84 of file Rmath.h.

Referenced by Rcpp::internal::empty_data_frame().

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

Definition at line 48 of file Rmath.h.

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

Definition at line 117 of file Rmath.h.

5.1.1.19 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 double R::digamma (double *x*) [inline]

Definition at line 197 of file Rmath.h.

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

Definition at line 66 of file Rmath.h.

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

Definition at line 154 of file Rmath.h.

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

Definition at line 129 of file Rmath.h.

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

Definition at line 78 of file Rmath.h.

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

Definition at line 35 of file Rmath.h.

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

Definition at line 171 of file Rmath.h.

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

Definition at line 142 of file Rmath.h.

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

Definition at line 186 of file Rmath.h.

5.1.1.33 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(), Rcpp::Datetime::getYearday(), and Rcpp::Datetime↩ ::update_tm().

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

Definition at line 42 of file Rmath.h.

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

Definition at line 148 of file Rmath.h.

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

Definition at line 180 of file Rmath.h.

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

Definition at line 32 of file Rmath.h.

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

5.1.1.38 double R::expm1 (double *x*) [inline]

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

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

Definition at line 231 of file Rmath.h.

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

Definition at line 232 of file Rmath.h.

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

Definition at line 234 of file Rmath.h.

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

Definition at line 235 of file Rmath.h.

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

Definition at line 236 of file Rmath.h.

5.1.1.44 double R::ftrunc (double *x*) [inline]

Definition at line 237 of file Rmath.h.

5.1.1.45 double R::gammafn (double *x*) [inline]

Definition at line 192 of file Rmath.h.

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

Definition at line 220 of file Rmath.h.

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

Definition at line 229 of file Rmath.h.

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

Definition at line 230 of file Rmath.h.

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

Definition at line 203 of file Rmath.h.

Referenced by Rcpp::internal::lfactorial().

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

Definition at line 206 of file Rmath.h.

Referenced by Rcpp::internal::lfactorial().

5.1.1.51 double R::lgamma1p (double *a*) [inline]

Definition at line 55 of file Rmath.h.

5.1.1.52 double R::lgammafn (double *x*) [inline]

Definition at line 193 of file Rmath.h.

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

Definition at line 194 of file Rmath.h.

5.1.1.54 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(), pythag(), Rcpp::stats::qlogis_0(), and Rcpp::stats::qlogis_1().

5.1.1.55 double R::log1pexp (double *x*) [inline]

Definition at line 54 of file Rmath.h.

5.1.1.56 double R::log1pmx (double *x*) [inline]

Definition at line 53 of file Rmath.h.

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

Definition at line 56 of file Rmath.h.

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

Definition at line 57 of file Rmath.h.

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

Definition at line 30 of file Rmath.h.

Referenced by Rcpp::stats::TGenerator::operator()(), Rcpp::stats::LNormGenerator::operator()(), Rcpp::stats::Norm↩ Generator::operator()(), Rcpp::stats::NormGenerator__sd1::operator()(), Rcpp::stats::LNormGenerator_1::operator()(), Rcpp::stats::NormGenerator__mean0::operator()(), Rcpp::stats::LNormGenerator_0::operator()(), and Rcpp::stats::↩ NormGenerator__mean0__sd1::operator()().

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

Definition at line 73 of file Rmath.h.

5.1.1.64 double R::pentagamma (double *x*) [inline]

Definition at line 200 of file Rmath.h.

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

Definition at line 112 of file Rmath.h.

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

Definition at line 118 of file Rmath.h.

5.1.1.69 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 double R::plnorm (double *x*, double *ml*, double *sl*, int *lt*, int *lg*) [inline]

Definition at line 67 of file Rmath.h.

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

Definition at line 79 of file Rmath.h.

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

Definition at line 172 of file Rmath.h.

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

Definition at line 143 of file Rmath.h.

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

Definition at line 196 of file Rmath.h.

Referenced by Rcpp::internal::lfactorial().

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

Definition at line 187 of file Rmath.h.

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

Definition at line 91 of file Rmath.h.

5.1.1.84 double R::ptukey (double *q*, double *rr*, double *cc*, double *df*, int *lt*, int *lg*) [inline]

Definition at line 176 of file Rmath.h.

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

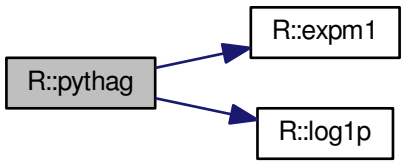
Definition at line 181 of file Rmath.h.

5.1.1.88 double R::pythag (double *a*, double *b*) [inline]

Definition at line 222 of file Rmath.h.

References expm1(), and log1p().

Here is the call graph for this function:



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

Definition at line 62 of file Rmath.h.

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

Definition at line 98 of file Rmath.h.

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

Definition at line 107 of file Rmath.h.

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

Definition at line 74 of file Rmath.h.

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

Definition at line 113 of file Rmath.h.

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

Definition at line 86 of file Rmath.h.

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

Definition at line 50 of file Rmath.h.

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

Definition at line 119 of file Rmath.h.

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

Definition at line 125 of file Rmath.h.

5.1.1.98 double R::qlnorm (double *p*, double *ml*, double *sl*, int *lt*, int *lg*) [inline]

Definition at line 68 of file Rmath.h.

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

Definition at line 156 of file Rmath.h.

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

Definition at line 162 of file Rmath.h.

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

Definition at line 131 of file Rmath.h.

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

Definition at line 137 of file Rmath.h.

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

Definition at line 80 of file Rmath.h.

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

Definition at line 168 of file Rmath.h.

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

Definition at line 37 of file Rmath.h.

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

Definition at line 173 of file Rmath.h.

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

Definition at line 144 of file Rmath.h.

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

Definition at line 188 of file Rmath.h.

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

Definition at line 92 of file Rmath.h.

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

Definition at line 177 of file Rmath.h.

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

Definition at line 44 of file Rmath.h.

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

Definition at line 150 of file Rmath.h.

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

Definition at line 182 of file Rmath.h.

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

Definition at line 63 of file Rmath.h.

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

Definition at line 99 of file Rmath.h.

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

Definition at line 108 of file Rmath.h.

5.1.1.117 double R::rchisq (double *df*) [inline]

Definition at line 75 of file Rmath.h.

5.1.1.118 double R::rexp (double *sl*) [inline]

Definition at line 114 of file Rmath.h.

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

Definition at line 87 of file Rmath.h.

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

Definition at line 51 of file Rmath.h.

5.1.1.121 double R::rgeom (double *p*) [inline]

Definition at line 120 of file Rmath.h.

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

Definition at line 126 of file Rmath.h.

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

Definition at line 69 of file Rmath.h.

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

Definition at line 157 of file Rmath.h.

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

Definition at line 102 of file Rmath.h.

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

Definition at line 163 of file Rmath.h.

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

Definition at line 132 of file Rmath.h.

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

Definition at line 81 of file Rmath.h.

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

Definition at line 38 of file Rmath.h.

5.1.1.130 double R::rpois (double *mu*) [inline]

Definition at line 145 of file Rmath.h.

5.1.1.131 double R::rsignrank (double *n*) [inline]

Definition at line 189 of file Rmath.h.

5.1.1.132 double R::rt (double *n*) [inline]

Definition at line 93 of file Rmath.h.

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

Definition at line 45 of file Rmath.h.

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

Definition at line 151 of file Rmath.h.

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

Definition at line 183 of file Rmath.h.

5.1.1.136 double R::sign (double *x*) [inline]

Definition at line 233 of file Rmath.h.

5.1.1.137 double R::tetragamma (double *x*) [inline]

Definition at line 199 of file Rmath.h.

5.1.1.138 double R::trigamma (double *x*) [inline]

Definition at line 198 of file Rmath.h.

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

Definition at line 31 of file Rmath.h.

Referenced by Rcpp::stats::CauchyGenerator::operator()(), Rcpp::stats::LogisGenerator::operator()(), Rcpp::stats::WeibullGenerator::operator()(), Rcpp::stats::UnifGenerator::operator()(), Rcpp::stats::CauchyGenerator_1::operator()(), Rcpp::stats::WeibullGenerator__scale1::operator()(), Rcpp::stats::UnifGenerator__0__1::operator()(), Rcpp::stats::LogisGenerator_1::operator()(), Rcpp::stats::CauchyGenerator_0::operator()(), and Rcpp::stats::LogisGenerator_0::operator()().

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 Argument
- class Armor
- class AttributeProxyPolicy
- class binary_call
- class BindingPolicy
- struct can_have_na
- class ChildVector
- class class_Base
- class ConstInputParameter
- class ConstMatrixColumn
- class ConstMatrixRow
- class ConstReferenceInputParameter
- class CppClass
- class CppFinalizer
- class CppFunction
- class CppFunctionBase
- class CppInheritedMethod
- class CppInheritedProperty

- class CppMethod
- class CppObject
- class CppProperty
- class DataFrame_Impl
- class Date
- class Datetime
- class DatetimeVector
- class DateVector
- class Dimension
- class DottedPairImpl
- class DottedPairProxyPolicy
- class enum_
- class exception
- class Fast
- class FieldProxyPolicy
- class file_exists
- class file_io_error
- class file_not_found
- class fixed_call
- class FunctionFinalizer
- class Generator
- struct GenericProxy
- class InputParameter
- class ListOf
- struct lsinfo
- class Matrix
- class MatrixBase
- class MatrixColumn
- class MatrixRow
- class Module
- class Na_Proxy
- class NamesProxyPolicy
- class no_init_matrix
- class no_init_vector
- class no_such_env
- class NoProtectStorage
- class Nullable
- class object
- class PreserveStorage
- class ProtectedProxyPolicy
- class Range
- class ReferenceInputParameter
- class result
- class RNGScope
- class RObjectMethods
- class Rostream
- class Rstreambuf
- struct rule
- class S4_CppConstructor
- class S4_CppOverloadedMethods
- class S4_field

- class Shelter
- class Shield
- class SignedConstructor
- class SignedFactory
- class SignedMethod
- class SingleLogicalResult
- class SlotProxyPolicy
- struct state
- class String
- class StringTransformer
- class SubMatrix
- class SubsetProxy
- class TagProxyPolicy
- class Timer
- struct ttinfo
- struct tzhead
- class unary_call
- class Vector
- class VectorBase
- struct void_type
- class XPtr

Typedefs

- typedef uint64_t nanotime_t
- typedef DataFrame_Impl< PreserveStorage > DataFrame
- 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)
- 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)
- template<>
  SEXP wrap< Rcpp::Datetime > (const Rcpp::Datetime &dt)
- Datetime operator+ (const Datetime &datetime, double 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)
- 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
- 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 bindingIsLocked (const std::string &name) const
- bool bindingIsActive (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)
- void update (SEXP)
- Environment new_env (int size=29)
- Environment new_env (SEXP parent, int size=29)
- static std::string toString (const int i)
- void warning (const std::string &message)
- 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)
- void NORET stop (const std::string &message)
- 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)
- 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_lcons (SEXP car, SEXP cdr)
- SEXP Rcpp_list2 (SEXP x0, SEXP x1)
- SEXP Rcpp_lang2 (SEXP x0, SEXP x1)
- SEXP Rcpp_list3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp_lang3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp_list4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP Rcpp_lang4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP Rcpp_list5 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4)
- SEXP Rcpp_lang5 (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_lang6 (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)
- SEXP Rcpp_lang9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)
- SEXP Rcpp_list10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)
- SEXP Rcpp_lang10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)
- SEXP 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_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_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_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_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_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_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_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_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_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_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_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_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_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_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_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_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_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_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_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)
- RCPP_API_CLASS (Language_Impl)
- Language_Impl ()
- Language_Impl (SEXP x)
- 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)
- 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)
- 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 rnbinom (int n, double siz, double prob)
- NumericVector rnbinom_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< R↩ TYPE >::type lhs, const Rcpp::VectorBase< RTYPE, NA, T > &vec, typename Rcpp::traits::storage_type< R↩ TYPE >::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::Vector↩ Base< 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::Vector↩
  Base< 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::Vector↩
  Base< 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< DatetimeVector > is_na (const DatetimeVector &x)
- sugar::IsNa_Vector_is_na< DateVector > is_na (const DateVector &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, N↩
  A_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< R↩
  TYPE >::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< RT↩
  YPE >::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, EXPONEN↩
  T_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)
- 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_result_of< Function, T >::type >::rtype >::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)
- 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)
- 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 LHS_NA, typename LHS_T >
  sugar::LowerTri< RTYPE, LHS_NA, LHS_T > lower_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_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 LHS_NA, typename LHS_T >
  sugar::UpperTri< RTYPE, LHS_NA, LHS_T > upper_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_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)

- 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 > tranpose_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 (int 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)
- 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, long *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 time_t transtime (const time_t janfirst, const int year, const struct rule *const rulep, const long offset)
- static struct tm * timesub (const time_t *const timep, const long offset, const struct state *const sp, struct tm *const tmp)
- static int leaps_thru_end_of (const int y)
- static int increment_overflow (int *number, int delta)
- static long detzcode (const char *const codep)
- static time_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, long *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 long 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
- static Rostream< false > Rcerr
- 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 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 typedef Matrix<STRSXP> Rcpp::CharacterMatrix

Definition at line 46 of file instantiation.h.

5.2.2.2 typedef Vector<STRSXP> Rcpp::CharacterVector

Definition at line 34 of file instantiation.h.

5.2.2.3 typedef Matrix<CPLXSXP> Rcpp::ComplexMatrix

Definition at line 40 of file instantiation.h.

5.2.2.4 typedef Vector<CPLXSXP> Rcpp::ComplexVector

Definition at line 27 of file instantiation.h.

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

Definition at line 39 of file Language.h.

5.2.2.6 typedef DataFrame_Impl<PreserveStorage> Rcpp::DataFrame

Definition at line 136 of file DataFrame.h.

5.2.2.7 typedef DottedPair_Impl<PreserveStorage> Rcpp::DottedPair

Definition at line 45 of file DottedPair.h.

5.2.2.8 typedef Vector<REALSXP> Rcpp::DoubleVector

Definition at line 31 of file instantiation.h.

5.2.2.9 typedef Environment_Impl<PreserveStorage> Rcpp::Environment

Definition at line 399 of file Environment.h.

5.2.2.10 typedef Matrix<EXPRSXP> Rcpp::ExpressionMatrix

Definition at line 50 of file instantiation.h.

5.2.2.11 typedef Vector<EXPRSXP> Rcpp::ExpressionVector

Definition at line 38 of file instantiation.h.

5.2.2.12 typedef Formula_Impl<PreserveStorage> Rcpp::Formula

Definition at line 75 of file Formula.h.

5.2.2.13 typedef Function_Impl<PreserveStorage> Rcpp::Function

Examples:

SugarPerformance/Timer.h.

Definition at line 111 of file Function.h.

5.2.2.14 typedef Matrix<VECSXP> Rcpp::GenericMatrix

Definition at line 48 of file instantiation.h.

5.2.2.15 typedef Vector<VECSXP> Rcpp::GenericVector

Definition at line 36 of file instantiation.h.

5.2.2.16 typedef Matrix<INTSXP> Rcpp::IntegerMatrix

Definition at line 41 of file instantiation.h.

5.2.2.17 typedef Vector<INTSXP> Rcpp::IntegerVector

Definition at line 28 of file instantiation.h.

5.2.2.18 typedef InternalFunction_Impl<PreserveStorage> Rcpp::InternalFunction

Definition at line 62 of file InternalFunction.h.

5.2.2.19 typedef Language_Impl<PreserveStorage> Rcpp::Language

Definition at line 159 of file Language.h.

5.2.2.20 typedef Vector<VECSXP> Rcpp::List

Definition at line 37 of file instantiation.h.

5.2.2.21 typedef Matrix<VECSXP> Rcpp::ListMatrix

Definition at line 49 of file instantiation.h.

5.2.2.22 typedef Matrix<LGLSXP> Rcpp::LogicalMatrix

Definition at line 42 of file instantiation.h.

5.2.2.23 typedef Vector<LGLSXP> Rcpp::LogicalVector

Definition at line 29 of file instantiation.h.

5.2.2.24 typedef uint64_t Rcpp::nanotime_t

Definition at line 47 of file Timer.h.

5.2.2.25 typedef Matrix<REALSXP> Rcpp::NumericMatrix

Definition at line 43 of file instantiation.h.

5.2.2.26 typedef Vector<REALSXP> Rcpp::NumericVector

Definition at line 30 of file instantiation.h.

5.2.2.27 typedef Pairlist_Impl<PreserveStorage> Rcpp::Pairlist

Definition at line 51 of file Pairlist.h.

5.2.2.28 typedef Promise_Impl<PreserveStorage> Rcpp::Promise

Definition at line 73 of file Promise.h.

5.2.2.29 typedef Matrix<RAWSXP> Rcpp::RawMatrix

Definition at line 44 of file instantiation.h.

5.2.2.30 typedef Vector<RAWSXP> Rcpp::RawVector

Definition at line 32 of file instantiation.h.

5.2.2.31 typedef Reference_Impl<PreserveStorage> Rcpp::Reference

Definition at line 70 of file Reference.h.

5.2.2.32 typedef RObject_Impl<PreserveStorage> Rcpp::RObject

Definition at line 56 of file RObject.h.

5.2.2.33 typedef S4_Impl<PreserveStorage> Rcpp::S4

Definition at line 72 of file S4.h.

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

Definition at line 81 of file StretchyList.h.

5.2.2.35 typedef Matrix<STRSXP> Rcpp::StringMatrix

Definition at line 47 of file instantiation.h.

5.2.2.36 typedef Vector<STRSXP> Rcpp::StringVector

Definition at line 35 of file instantiation.h.

5.2.2.37 typedef Symbol_Impl<NoProtectStorage> Rcpp::Symbol

Definition at line 80 of file Symbol.h.

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

Definition at line 139 of file Module.h.

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

Definition at line 140 of file Module.h.

5.2.2.40 typedef WeakReference_Impl<PreserveStorage> Rcpp::WeakReference

Definition at line 62 of file WeakReference.h.

5.2.3 Function Documentation

5.2.3.1 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 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 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 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 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.

5.2.3.6 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 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 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 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 143 of file as.h.

Referenced by CppProperty_GetMethod_SetMethod< Class, PROP >::set(), CppProperty_GetConstMethod_Set↩Method< Class, PROP >::set(), CppProperty_GetMethod_SetPointer< Class, PROP >::set(), CppProperty_Get↩ConstMethod_SetPointer< Class, PROP >::set(), CppProperty_GetPointer_SetMethod< Class, PROP >::set(), Cpp↩Property_GetPointer_SetPointer< Class, PROP >::set(), and Rcpp::attributes::SourceFileAttributesParser::Source↩FileAttributesParser().

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

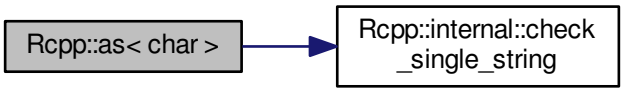
Definition at line 156 of file as.h.

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

Definition at line 147 of file as.h.

References Rcpp::internal::check_single_string().

Here is the call graph for this function:

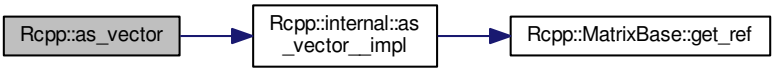


5.2.3.12 template<int RTYPE, bool NA, typename T > Rcpp::Vector<RTYPE> Rcpp::as_vector (const MatrixBase< RTYPE, NA, T > & *t*) [inline]

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

Attempts to assign x to name in this environment

Parameters

name	name of the object to assign
x	object to assign

Returns

true if the assign was successfull see ?bindingIsLocked

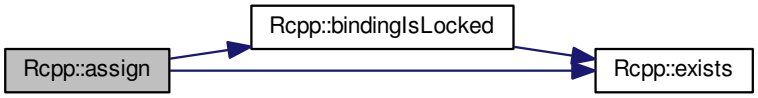
Exceptions

binding_is_locked	if the binding is locked

Definition at line 209 of file Environment.h.

References bindingIsLocked(), and exists().

Here is the call graph for this function:



5.2.3.14 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

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

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

Definition at line 152 of file as.h.

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

Returns

the base environment. See ?baseenv

Definition at line 345 of file Environment.h.

Referenced by Rcpp::attributes::SourceFileAttributesParser::attributeWarning(), Rcpp::attributes::createDirectory(), Rcpp::attributes::isRoxygenCpp(), Rcpp::attributes::removeFile(), and Rcpp::attributes::showWarning().

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

Returns

the base namespace. See ?baseenv

Definition at line 352 of file Environment.h.

5.2.3.18 bool Rcpp::bindingIsActive (const std::string & *name*) const

Parameters

name	name of a binding

Returns

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

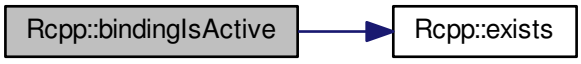
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 314 of file Environment.h.

References exists().

Here is the call graph for this function:



5.2.3.19 bool Rcpp::bindingIsLocked (const std::string & *name*) const

Parameters

name	name of a potential binding

Returns

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

Exceptions

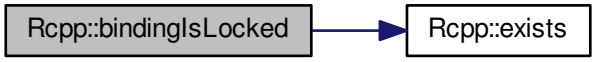
no_such_binding	if there is no such binding in this environment

Definition at line 299 of file Environment.h.

References exists().

Referenced by assign(), and remove().

Here is the call graph for this function:



5.2.3.20 void Rcpp::checkUserInterrupt ( ) [inline]

Definition at line 59 of file Interrupt.h.

5.2.3.21 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(), Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs, and Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs.

Here is the call graph for this function:

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

Definition at line 33 of file clone.h.

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

5.2.3.23 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]

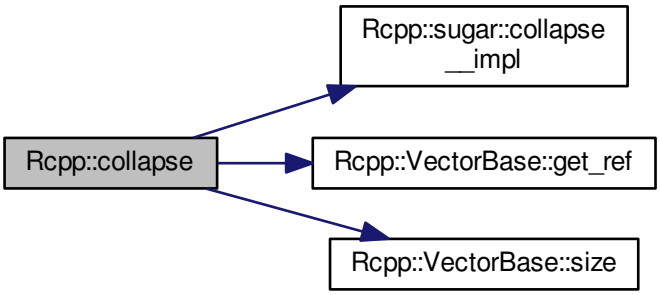
Definition at line 55 of file col.h.

5.2.3.24 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::Vector↩Base< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



5.2.3.25 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.

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

Definition at line 214 of file Matrix.h.

5.2.3.27 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.

5.2.3.28 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.

5.2.3.29 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.

5.2.3.30 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.

5.2.3.31 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.

5.2.3.32 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.

5.2.3.33 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.

5.2.3.34 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.

5.2.3.35 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.

5.2.3.36 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.

5.2.3.37 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.

5.2.3.38 static long Rcpp::detzcode (const char *const *codep*) [static]

Definition at line 398 of file Date.cpp.

Referenced by tzload().

5.2.3.39 static time_t Rcpp::detzcode64 (const char *const *codep*) [static]

Definition at line 408 of file Date.cpp.

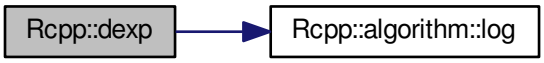
Referenced by tzload().

5.2.3.40 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:



5.2.3.41 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 lower_tri(), and upper_tri().

5.2.3.42 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.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs.

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

5.2.3.43 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.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs.

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

Definition at line 418 of file Date.cpp.

References SECSPERREPEAT, SECSPERREPEAT_BITS, TYPE_BIT, TYPE_INTEGRAL, and TYPE_SIGNED.

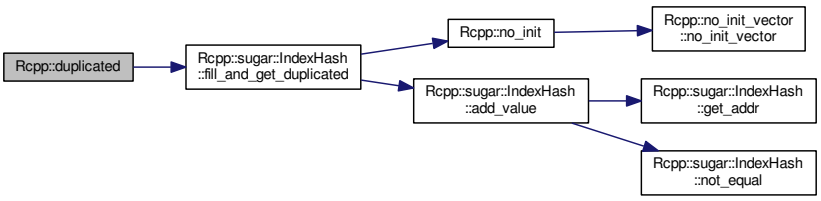
Referenced by tzload().

5.2.3.45 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:



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

Returns

The empty environment. See ?emptyenv

Definition at line 338 of file Environment.h.

5.2.3.47 Rcpp::Environment_Impl (SEXP *x*)

wraps the given environment

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

Definition at line 59 of file Environment.h.

Referenced by namespace_env(), new_child(), and parent().

5.2.3.48 Rcpp::Environment_Impl::Environment_Impl (const std::string & *name*)

Gets the environment associated with the given name

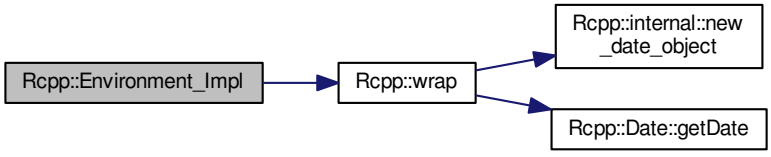
Parameters

name	name of the environment, e.g "package:Rcpp"

Definition at line 34 of file Environment.h.

References wrap().

Here is the call graph for this function:



5.2.3.49 Rcpp::Environment_Impl::Environment_Impl (int *pos*)

Gets the environment in the given position of the search path

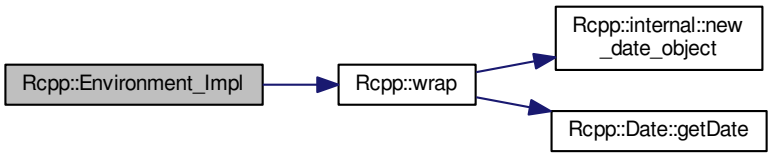
Parameters

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

Definition at line 41 of file Environment.h.

References wrap().

Here is the call graph for this function:



5.2.3.50 SEXP Rcpp::eval ( ) const

eval this call in the global environment

Definition at line 136 of file Language.h.

References Rcpp_eval().

Here is the call graph for this function:

5.2.3.51 SEXP Rcpp::eval (SEXP *env*) const

eval this call in the requested environment

Definition at line 143 of file Language.h.

References Rcpp_eval().

Here is the call graph for this function:

5.2.3.52 bool Rcpp::exists (const std::string & *name*) const

Indicates if an object called name exists in the environment

Parameters

name	name of the object

Returns

true if the object exists in the environment

Definition at line 192 of file Environment.h.

Referenced by assign(), bindingIsActive(), bindingIsLocked(), Rcpp::attributes::ExportsGenerator::ExportsGenerator(), lockBinding(), remove(), and unlockBinding().

5.2.3.53 SEXP Rcpp::fast_eval ( ) const

Definition at line 147 of file Language.h.

5.2.3.54 SEXP Rcpp::fast_eval (SEXP *env*) const

Definition at line 150 of file Language.h.

References update().

Here is the call graph for this function:

5.2.3.55 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.

5.2.3.56 SEXP Rcpp::find (const std::string & *name*) const

Get an object from the environment or one of its parents

Parameters

name	name of the object

Definition at line 147 of file Environment.h.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), and Rcpp::attributes::SourceFileAttributesParser::SourceFile↩AttributesParser().

5.2.3.57 SEXP Rcpp::find (Symbol *name*) const

Get an object from the environment or one of its parents

Parameters

name	symbol name to call

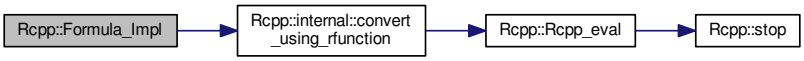
Definition at line 167 of file Environment.h.

5.2.3.58 Rcpp::Formula_Impl (SEXP *x*)

Definition at line 40 of file Formula.h.

References Rcpp::internal::convert_using_rfunction().

Here is the call graph for this function:

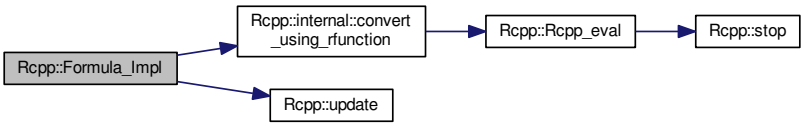


5.2.3.59 Rcpp::Formula_Impl (const std::string & *code*) [explicit]

Definition at line 69 of file Formula.h.

References Rcpp::internal::convert_using_rfunction(), and update().

Here is the call graph for this function:



5.2.3.60 SEXP Rcpp::get (const std::string & *name*) const

Get an object from the environment

Parameters

name	name of the object

Returns

a SEXP (possibly R_NilValue)

Definition at line 105 of file Environment.h.

Referenced by Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[]().

5.2.3.61 SEXP Rcpp::get (Symbol *name*) const

Get an object from the environment

Parameters

name	symbol name to call

Returns

a SEXP (possibly R_NilValue)

Definition at line 126 of file Environment.h.

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

Definition at line 42 of file get_return_type.h.

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

Definition at line 74 of file get_return_type.h.

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

Definition at line 66 of file get_return_type.h.

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

Definition at line 54 of file get_return_type.h.

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

Definition at line 70 of file get_return_type.h.

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

Definition at line 58 of file get_return_type.h.

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

Definition at line 62 of file get_return_type.h.

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

Definition at line 50 of file get_return_type.h.

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

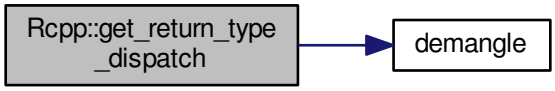
Definition at line 46 of file get_return_type.h.

5.2.3.71 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:

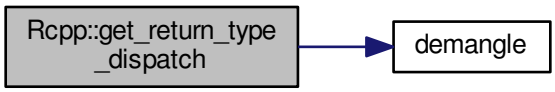


5.2.3.72 template<typename RESULT_TYPE > std::string Rcpp::get_return_type_dispatch (Rcpp::traits::true_type) [inline]

Definition at line 34 of file get_return_type.h.

References demangle().

Here is the call graph for this function:



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

Definition at line 490 of file Date.cpp.

References is_digit.

Referenced by getrule(), and getsecs().

5.2.3.74 static const char* Rcpp::getoffset (const char * *strp,* long *const *offsetp*) [static]

Definition at line 443 of file Date.cpp.

References getsecs().

Referenced by tzparse().

Here is the call graph for this function:

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

Definition at line 435 of file Date.cpp.

Referenced by tzparse().

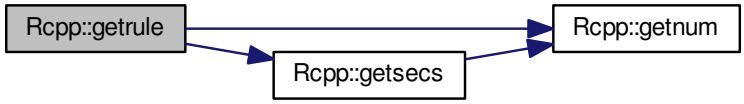
5.2.3.76 static const char * Rcpp::getrule (const char * *strp,* struct rule *const *rulep*) [static]

Definition at line 509 of file Date.cpp.

References DAY_OF_YEAR, 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:



5.2.3.77 static const char * Rcpp::getsecs (const char * *strp,* long * *secsp*) [static]

Definition at line 459 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:

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

Definition at line 426 of file Date.cpp.

References is_digit.

Referenced by tzparse().

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

Returns

the global environment. See ?globalenv

Definition at line 331 of file Environment.h.

5.2.3.80 struct tm * Rcpp::gmtime_ (const time_t *const *x*)

Definition at line 110 of file routines.h.

References GET_CALLABLE, and tm.

Referenced by gmtsub(), operator+(), registerFunctions(), Rcpp::Datetime::update_tm(), and Rcpp::Date::update_tm().

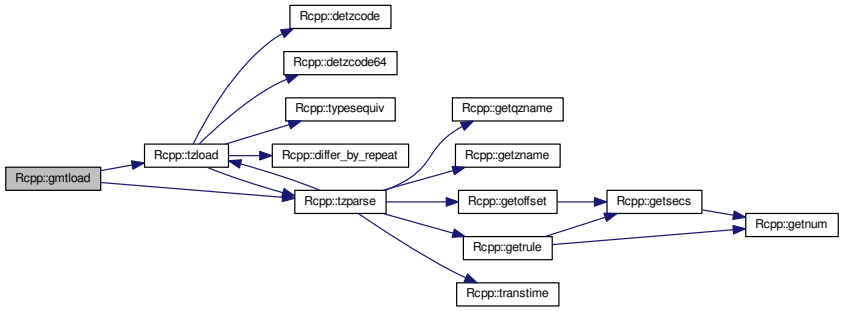
5.2.3.81 static void Rcpp::gmtload (struct state *const *sp*) [static]

Definition at line 1256 of file Date.cpp.

References tzload(), and tzparse().

Referenced by gmtsub().

Here is the call graph for this function:

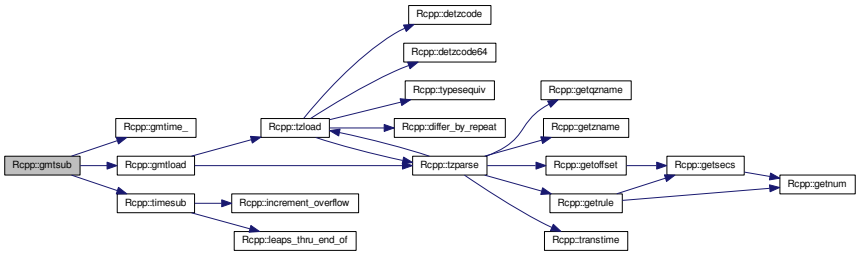


5.2.3.82 static struct tm* Rcpp::gmtsub (const time_t *const *timep*, const long *offset*, struct tm *const *tmp*) [static]

Definition at line 1261 of file Date.cpp.

References gmtime_(), gmtload(), gmtptr, and timesub().

Here is the call graph for this function:



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

Definition at line 34 of file grow.h.

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

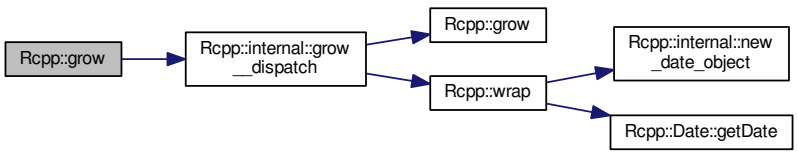
5.2.3.84 template<typename T > SEXP Rcpp::grow (const T & *head*, SEXP *tail*)

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().

Here is the call graph for this function:



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

Definition at line 68 of file grow.h.

References grow().

Here is the call graph for this function:

5.2.3.86 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.

References Rcpp::sugar::Head< RTYPE, NA, T >::n.

5.2.3.87 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.

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.

5.2.3.88 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.

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.

5.2.3.89 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.

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.

5.2.3.90 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.

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.

5.2.3.91 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.

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.

5.2.3.92 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.

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.

5.2.3.93 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.

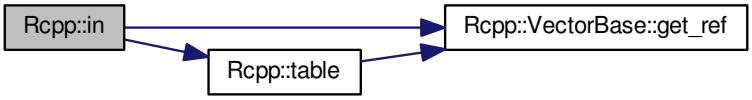
References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

5.2.3.94 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::VectorBase< RTYPE, na, VECTOR >::get_ref(), and table().

Here is the call graph for this function:



5.2.3.95 static int Rcpp::increment_overflow (int * *number,* int *delta*) [static]

Definition at line 390 of file Date.cpp.

Referenced by timesub().

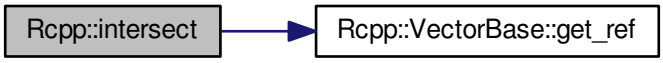
5.2.3.96 template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T > Vector<RTYPE> Rcpp::intersect (const VectorBase< RTYPE, LHS_NA, LHS_T > & *lhs,* const VectorBase< RTYPE, RHS_NA, RHS_T > & *rhs*) [inline]

Definition at line 171 of file setdiff.h.

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

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().

Here is the call graph for this function:



5.2.3.97 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 RCPP_API_CLASS().

5.2.3.98 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.

5.2.3.99 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().

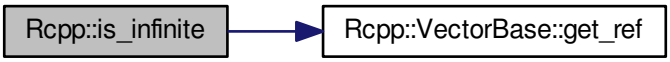
Here is the call graph for this function:

5.2.3.100 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:



5.2.3.101 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:

5.2.3.102 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 87 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()().

5.2.3.103 sugar::IsNa_Vector_is_na<DatetimeVector> Rcpp::is_na (const DatetimeVector & *x*) [inline]

Definition at line 91 of file is_na.h.

5.2.3.104 sugar::IsNa_Vector_is_na<DateVector> Rcpp::is_na (const DateVector & *x*) [inline]

Definition at line 94 of file is_na.h.

5.2.3.105 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:

5.2.3.106 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.

Referenced by Rcpp::sugar::SingleLogicalResult< false, Any< false, T > >::operator bool().

5.2.3.107 bool Rcpp::is_user_database ( ) const

Indicates if this is a user defined database.

Definition at line 323 of file Environment.h.

Referenced by ls().

5.2.3.108 bool Rcpp::isLocked ( ) const

Returns

true if this environment is locked see ?environmentIsLocked for details of what this means

Definition at line 231 of file Environment.h.

5.2.3.109 Rcpp::Language_Impl ( )

Definition at line 43 of file Language.h.

5.2.3.110 Rcpp::Language_Impl (SEXP *x*)

Attempts to convert the SEXP to a call

Exceptions

not_compatible	if the SEXP could not be converted to a call using as.call

Definition at line 51 of file Language.h.

5.2.3.111 Rcpp::Language_Impl (const std::string & *symbol*) [explicit]

Creates a call using the given symbol as the function name

Parameters

symbol	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 64 of file Language.h.

5.2.3.112 Rcpp::Language_Impl (const Symbol & *symbol*) [explicit]

Creates a call using the given symbol as the function name

Parameters

symbol	symbol name to call

Language(Symbol("rnorm")) makes a SEXP similar to this:

```
call( "rnorm" )
```

Definition at line 76 of file Language.h.

5.2.3.113 Rcpp::Language_Impl (const Function & *function*) [explicit]

Creates a call to the function

Parameters

function	function to call

Definition at line 85 of file Language.h.

5.2.3.114 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.

References Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::fun.

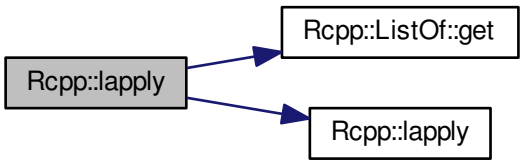
Referenced by lapply().

5.2.3.115 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:



5.2.3.116 static int Rcpp::leaps_thru_end_of (const int *y*) [static]

Definition at line 1130 of file Date.cpp.

Referenced by timesub().

5.2.3.117 void Rcpp::lock (bool *bindings* = false)

locks this environment. See ?lockEnvironment

Parameters

bindings	also lock the bindings of this environment ?

Definition at line 263 of file Environment.h.

5.2.3.118 void Rcpp::lockBinding (const std::string & *name*)

Locks the given binding in the environment. see ?bindingIsLocked

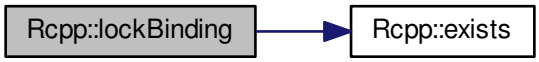
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 273 of file Environment.h.

References exists().

Here is the call graph for this function:

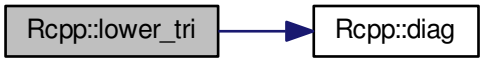


5.2.3.119 template<int RTYPE, bool LHS_NA, typename LHS_T > sugar::LowerTri<RTYPE,LHS_NA,LHS_T> Rcpp::lower_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > & *lhs,* bool *diag* = `false`) [inline]

Definition at line 75 of file lower_tri.h.

References diag().

Here is the call graph for this function:



5.2.3.120 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

all	same meaning as in ?ls

Definition at line 87 of file Environment.h.

References is_user_database().

Here is the call graph for this function:

5.2.3.121 template<typename UnaryOperator > StringTransformer<UnaryOperator> Rcpp::make_string_transformer (const UnaryOperator & *fun*)

Definition at line 47 of file StringTransformer.h.

5.2.3.122 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.

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.

5.2.3.123 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::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, and Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref().

Here is the call graph for this function:

5.2.3.124 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::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, and Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref().

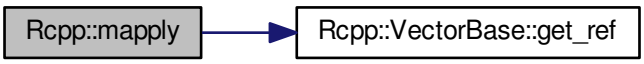
Here is the call graph for this function:

5.2.3.125 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::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, and Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref().

Here is the call graph for this function:



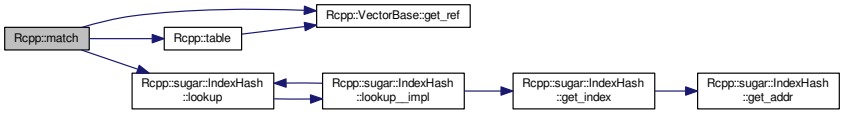
5.2.3.126 template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T > IntegerVector Rcpp::match (const VectorBase< RTYPE, NA, T > & *x,* const VectorBase< RTYPE, RHS_NA, RHS_T > & *table_*) [inline]

Definition at line 28 of file match.h.

References Rcpp::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(), and Rcpp::attributes::SourceFileAttributesParser::sourceDependencies().

Here is the call graph for this function:



5.2.3.127 template<int RTYPE, bool NA, typename T > sugar::Max<RTYPE,NA,T> Rcpp::max (const VectorBase< RTYPE, NA, T > & *x*)

Definition at line 81 of file max.h.

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

Referenced by Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE(), Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE(), and operator<<().

Here is the call graph for this function:

5.2.3.128 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(), and Rcpp::sugar::Var< CPLXSXP, NA, T >::get().

5.2.3.129 template<bool NA, typename T > sugar::Mean<INTSXP,NA,T> Rcpp::mean (const VectorBase< INTSXP, NA, T > & *t*) [inline]

Definition at line 145 of file mean.h.

5.2.3.130 template<bool NA, typename T > sugar::Mean<CPLXSXP,NA,T> Rcpp::mean (const VectorBase< CPLXSXP, NA, T > & *t*) [inline]

Definition at line 150 of file mean.h.

5.2.3.131 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.

5.2.3.132 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.

5.2.3.133 template<int RTYPE, bool NA, typename T > sugar::Min<RTYPE,NA,T> Rcpp::min (const VectorBase< RTYPE, NA, T > & *x*)

Definition at line 80 of file min.h.

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

Referenced by Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMax< RTYPE, false, T >::get(), Rcpp::sugar::WhichMin< RTYPE, false, T >::get(), Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE(), Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE(), and operator<<().

Here is the call graph for this function:

5.2.3.134 double Rcpp::mktime00 (struct tm & *tm*) [inline]

Definition at line 104 of file routines.h.

References GET_CALLABLE.

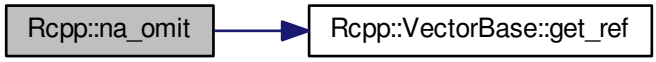
Referenced by Rcpp::Date::Date(), and registerFunctions().

5.2.3.135 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:



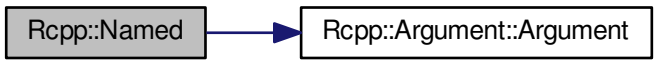
5.2.3.136 Argument Rcpp::Named (const std::string & *name*) [inline]

Definition at line 40 of file Named.h.

References Rcpp::Argument::Argument().

Referenced by Rcpp::attributes::createDirectory(), fastLm(), and Rcpp::attributes::showWarning().

Here is the call graph for this function:



5.2.3.137 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.

References Rcpp::Argument::name.

5.2.3.138 static Environment_Impl Rcpp::namespace_env (const std::string & *package*) [static]

Parameters

name	the name of the package of which we want the namespace

Returns

the namespace of the package

Exceptions

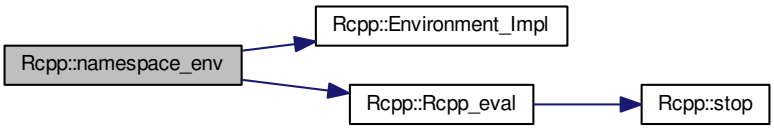
no_such_namespace	

Definition at line 370 of file Environment.h.

References Environment_Impl(), and Rcpp_eval().

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

Here is the call graph for this function:



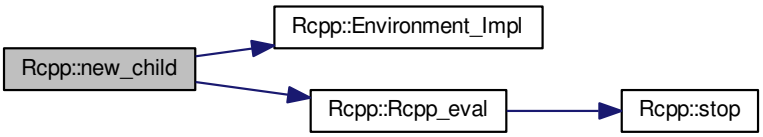
5.2.3.139 Environment_Impl Rcpp::new_child (bool *hashed*)

creates a new environment whose this is the parent

Definition at line 392 of file Environment.h.

References Environment_Impl(), and Rcpp_eval().

Here is the call graph for this function:

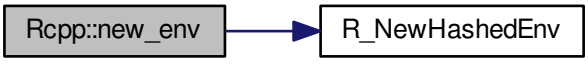


5.2.3.140 Environment Rcpp::new_env (int *size* = 29) [inline]

Definition at line 403 of file Environment.h.

References R_NewHashedEnv().

Here is the call graph for this function:

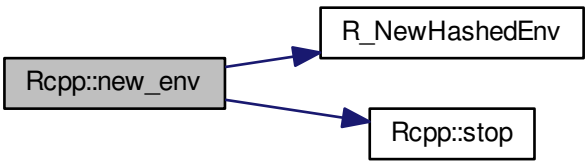


5.2.3.141 Environment Rcpp::new_env (SEXP *parent*, int *size* = 29) [inline]

Definition at line 408 of file Environment.h.

References R_NewHashedEnv(), and stop().

Here is the call graph for this function:



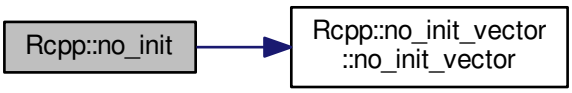
5.2.3.142 no_init_vector Rcpp::no_init (int *size*) [inline]

Definition at line 69 of file no_init.h.

References Rcpp::no_init_vector::no_init_vector().

Referenced by 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(), and Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector().

Here is the call graph for this function:



5.2.3.143 no_init_matrix Rcpp::no_init (int *nr,* int *nc*) [inline]

Definition at line 73 of file no_init.h.

5.2.3.144 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().

5.2.3.145 sugar::NonaPrimitive<double> Rcpp::noNA (double *x*) [inline]

Definition at line 78 of file nona.h.

5.2.3.146 sugar::NonaPrimitive<int> Rcpp::noNA (int *x*) [inline]

Definition at line 81 of file nona.h.

5.2.3.147 Rcpp::operator SEXP ( ) const [inline]

Definition at line 53 of file StretchyList.h.

5.2.3.148 bool Rcpp::operator!= (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 121 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.149 bool Rcpp::operator!= (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 150 of file Date.h.

References Rcpp::Date::m_d.

5.2.3.150 bool Rcpp::operator!= (const String::StringProxy & *lhs,* const String & *rhs*) [inline]

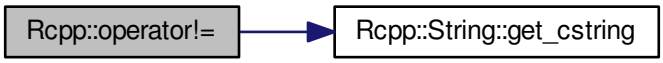
Definition at line 524 of file String.h.

5.2.3.151 bool Rcpp::operator!= (const String::const_StringProxy & *lhs,* const String & *rhs*) [inline]

Definition at line 532 of file String.h.

References Rcpp::String::get_cstring().

Here is the call graph for this function:



5.2.3.152 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.

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.

5.2.3.153 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.

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.

5.2.3.154 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.

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.

5.2.3.155 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.

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.

5.2.3.156 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.

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.

5.2.3.157 Datetime Rcpp::operator+ (const Datetime & *datetime,* double *offset*) [inline]

Definition at line 106 of file Datetime.h.

References gmtime_(), Rcpp::Datetime::m_dt, Rcpp::Datetime::m_tm, and Rcpp::Datetime::m_us.

Here is the call graph for this function:

5.2.3.158 Date Rcpp::operator+ (const Date & *date,* int *offset*) [inline]

Definition at line 136 of file Date.h.

References gmtime_(), Rcpp::Date::m_d, and Rcpp::Date::m_tm.

Here is the call graph for this function:

5.2.3.159 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.

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.

5.2.3.160 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.

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.

5.2.3.161 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.

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.

5.2.3.162 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.

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.

5.2.3.163 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.

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.

5.2.3.164 double Rcpp::operator- (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 115 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.165 double Rcpp::operator- (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 144 of file Date.h.

References Rcpp::Date::m_d.

5.2.3.166 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.

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.

5.2.3.167 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.

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.

5.2.3.168 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.

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.

5.2.3.169 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.

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.

5.2.3.170 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.

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.

5.2.3.171 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.

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.

5.2.3.172 bool Rcpp::operator< (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 116 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.173 bool Rcpp::operator< (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 145 of file Date.h.

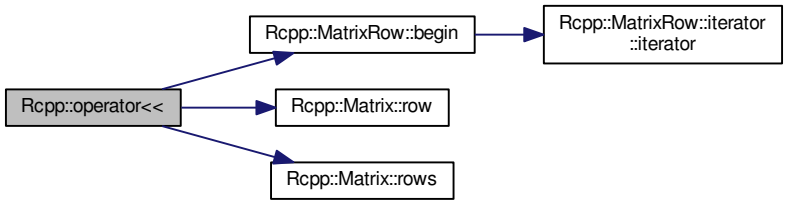
References Rcpp::Date::m_d.

5.2.3.174 template<template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Matrix< REALSXP, StoragePolicy > & *rhs*) [inline]

Definition at line 219 of file Matrix.h.

References Rcpp::MatrixRow< RTYPE >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::row(), and Rcpp::Matrix< RTYPE, StoragePolicy >::rows().

Here is the call graph for this function:

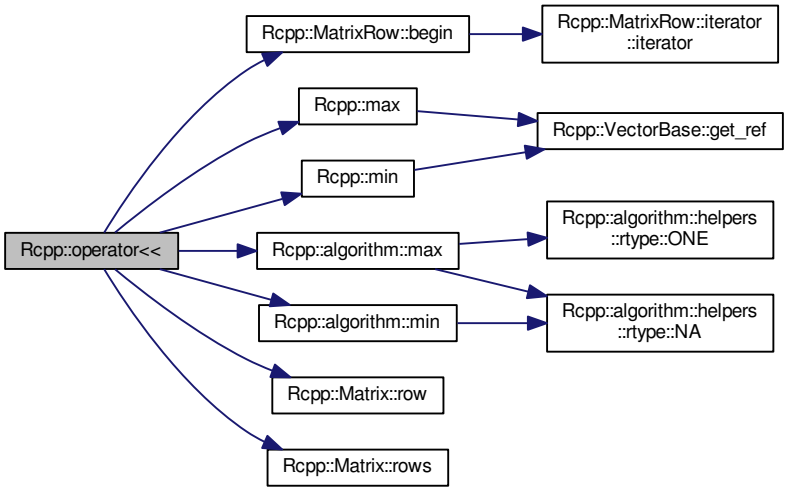


5.2.3.175 template<template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Matrix< INTSXP, StoragePolicy > & *rhs*) [inline]

Definition at line 286 of file Matrix.h.

References Rcpp::MatrixRow< RTYPE >::begin(), max(), Rcpp::algorithm::max(), min(), Rcpp::algorithm::min(), Rcpp::Matrix< RTYPE, StoragePolicy >::row(), and Rcpp::Matrix< RTYPE, StoragePolicy >::rows().

Here is the call graph for this function:

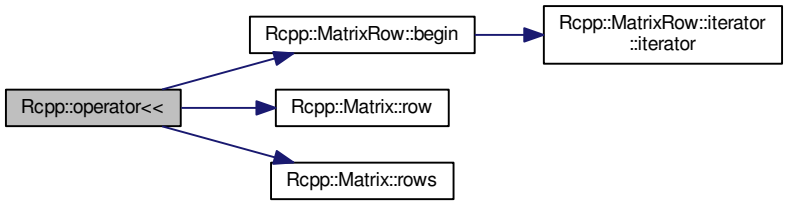


5.2.3.176 template<template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Matrix< STRSXP, StoragePolicy > & *rhs*) [inline]

Definition at line 352 of file Matrix.h.

References Rcpp::MatrixRow< RTYPE >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::row(), and Rcpp::Matrix< RTYPE, StoragePolicy >::rows().

Here is the call graph for this function:

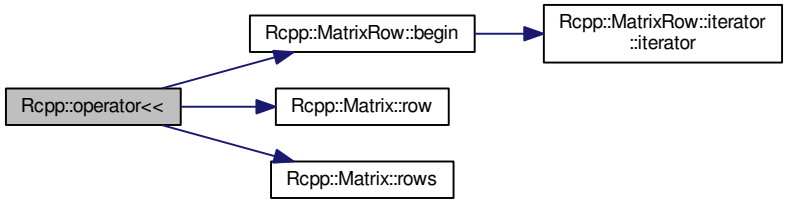


5.2.3.177 template<int RTYPE, template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Matrix< RTYPE, StoragePolicy > & *rhs*) [inline]

Definition at line 379 of file Matrix.h.

References Rcpp::MatrixRow< RTYPE >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::row(), and Rcpp::Matrix< RTYPE, StoragePolicy >::rows().

Here is the call graph for this function:

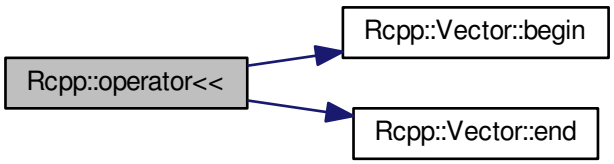


5.2.3.178 template<int RTYPE, template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Vector< RTYPE, StoragePolicy > & *rhs*) [inline]

Definition at line 1093 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:

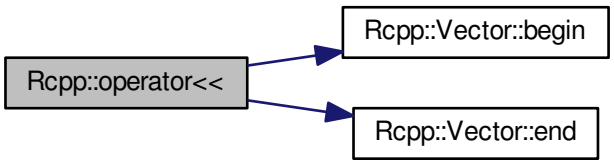


5.2.3.179 template<template< class > class StoragePolicy> std::ostream& Rcpp::operator<< (std::ostream & *s*, const Vector< STRSXP, StoragePolicy > & *rhs*) [inline]

Definition at line 1112 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



5.2.3.180 bool Rcpp::operator<= (const Datetime & *d1*, const Datetime & *d2*) [inline]

Definition at line 120 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.181 bool Rcpp::operator<= (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 149 of file Date.h.

References Rcpp::Date::m_d.

5.2.3.182 bool Rcpp::operator== (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 118 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.183 bool Rcpp::operator== (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 147 of file Date.h.

References Rcpp::Date::m_d.

Referenced by RCPP_API_CLASS().

5.2.3.184 bool Rcpp::operator== (const String::StringProxy & *lhs,* const String & *rhs*) [inline]

Definition at line 520 of file String.h.

5.2.3.185 bool Rcpp::operator== (const String::const_StringProxy & *lhs,* const String & *rhs*) [inline]

Definition at line 528 of file String.h.

5.2.3.186 bool Rcpp::operator> (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 117 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.187 bool Rcpp::operator> (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 146 of file Date.h.

References Rcpp::Date::m_d.

5.2.3.188 bool Rcpp::operator>= (const Datetime & *d1,* const Datetime & *d2*) [inline]

Definition at line 119 of file Datetime.h.

References Rcpp::Datetime::m_dt.

5.2.3.189 bool Rcpp::operator>= (const Date & *d1,* const Date & *d2*) [inline]

Definition at line 148 of file Date.h.

References Rcpp::Date::m_d.

5.2.3.190 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.

References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::fun, 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.

5.2.3.191 SEXP Rcpp::pairlist ( ) [inline]

Definition at line 30 of file grow.h.

Referenced by Rcpp::DottedPairImpl< CLASS >::replace().

5.2.3.192 Rcpp::Pairlist_Impl ( )

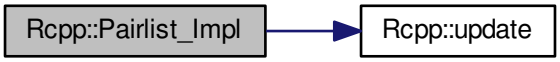
Definition at line 41 of file Pairlist.h.

5.2.3.193 Rcpp::Pairlist_Impl (SEXP *x*)

Definition at line 42 of file Pairlist.h.

References update().

Here is the call graph for this function:



5.2.3.194 Environment_Impl Rcpp::parent ( ) const

The parent environment of this environment

Definition at line 385 of file Environment.h.

References Environment_Impl().

Referenced by class_< Class >::derives().

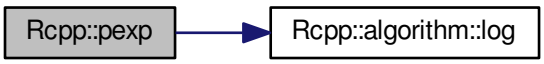
Here is the call graph for this function:

5.2.3.195 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:

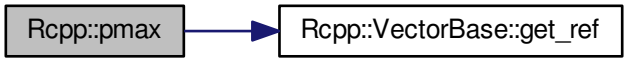


5.2.3.196 template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T > sugar::Pmax_Vector_↩ Vector<RTYPE,LHS_NA,LHS_T,RHS_NA,RHS_T> Rcpp::pmax (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & *lhs,* const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & *rhs*) [inline]

Definition at line 158 of file pmax.h.

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

Here is the call graph for this function:



5.2.3.197 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:

5.2.3.198 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().

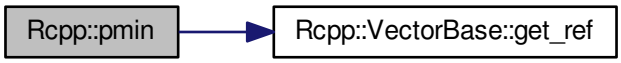
Here is the call graph for this function:

5.2.3.199 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:

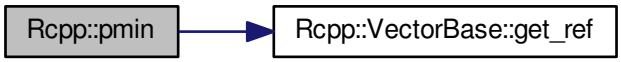


5.2.3.200 template<int RTYPE, bool LHS_NA, typename LHS_T > sugar::Pmin_Vector_Primitive<RTYPE,LHS_NA,LHS_T> Rcpp::pmin (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & *lhs,* typename Rcpp::traits::storage_type< RTYPE >::type *rhs*) [inline]

Definition at line 168 of file pmin.h.

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

Here is the call graph for this function:



5.2.3.201 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:

5.2.3.202 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:

5.2.3.203 void Rcpp::print (SEXP *s*) [inline]

Definition at line 25 of file print.h.

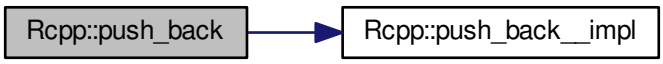
5.2.3.204 template<typename T > StretchyList_Impl& Rcpp::push_back (const T & *obj*) [inline]

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:



5.2.3.205 template<typename T > StretchyList_Impl& Rcpp::push_back__impl (const T & *obj,* traits::true_type) [private]

Referenced by push_back(), and push_front().

5.2.3.206 template<typename T > StretchyList_Impl& Rcpp::push_back__impl (const T & *obj,* traits::false_type) [private]

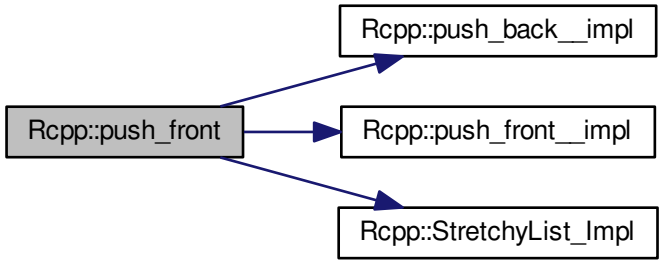
5.2.3.207 template<typename T > StretchyList_Impl& Rcpp::push_front (const T & *obj*) [inline]

Definition at line 63 of file StretchyList.h.

References push_back__impl(), push_front__impl(), and StretchyList_Impl().

Referenced by Rcpp::DottedPairImpl< CLASS >::insert().

Here is the call graph for this function:



5.2.3.208 template<typename T > StretchyList_Impl& Rcpp::push_front__impl (const T & *obj,* traits::true_type) [private]

Referenced by push_front().

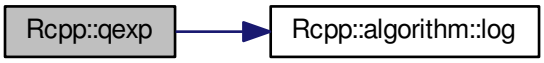
5.2.3.209 template<typename T > StretchyList_Impl& Rcpp::push_front__impl (const T & *obj,* traits::false_type) [private]

5.2.3.210 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:



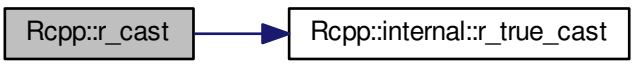
5.2.3.211 template<int TARGET> SEXP Rcpp::r_cast (SEXP *x*)

Definition at line 138 of file r_cast.h.

References Rcpp::internal::r_true_cast().

Referenced by Rcpp::internal::vector_from_string().

Here is the call graph for this function:



5.2.3.212 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().

Here is the call graph for this function:

5.2.3.213 NumericVector Rcpp::rbeta (int *n,* double *a,* double *b*) [inline]

Definition at line 106 of file random.h.

5.2.3.214 NumericVector Rcpp::rbinom (int *n,* double *nin,* double *pp*) [inline]

Definition at line 110 of file random.h.

5.2.3.215 NumericVector Rcpp::rcauchy (int *n,* double *location,* double *scale*) [inline]

Definition at line 114 of file random.h.

5.2.3.216 NumericVector Rcpp::rcauchy (int *n,* double *location*) [inline]

Definition at line 124 of file random.h.

5.2.3.217 NumericVector Rcpp::rcauchy (int *n*) [inline]

Definition at line 134 of file random.h.

5.2.3.218 NumericVector Rcpp::rchisq (int *n,* double *df*) [inline]

Definition at line 138 of file random.h.

5.2.3.219 Rcpp::RCPP_API_CLASS (DottedPair_Impl)

5.2.3.220 Rcpp::RCPP_API_CLASS (Symbol_Impl)

wraps the SEXP into a Symbol object.

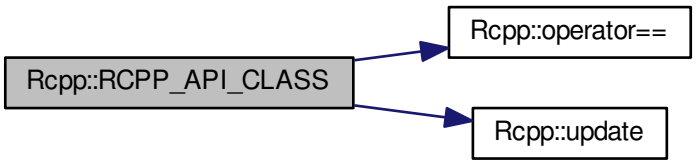
Parameters

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:



5.2.3.221 Rcpp::RCPP_API_CLASS (WeakReference_Impl)

wraps a weak reference

Parameters

x	presumably a SEXP of SEXTYPE WEAKREFSXP

Exceptions

not_compatible	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:

5.2.3.222 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:

5.2.3.223 Rcpp::RCPP_API_CLASS (RObject_Impl)

default constructor. uses R_NilValue

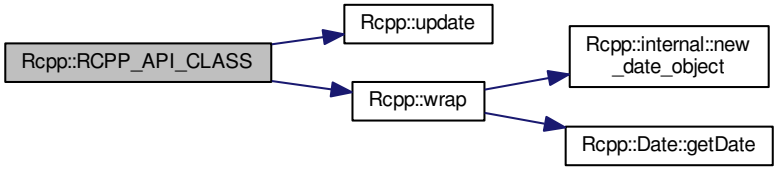
wraps a SEXP. The SEXP is automatically protected from garbage collection by this object and the protection vanishes when this object is destroyed

Assignement 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:



5.2.3.224 Rcpp::RCPP_API_CLASS (Reference_Impl)

S4 object (of a reference class)

5.2.3.225 Rcpp::RCPP_API_CLASS (Formula_Impl)

5.2.3.226 Rcpp::RCPP_API_CLASS (S4_Impl)

S4 object checks that x is an S4 object and wrap it.

Parameters

x	must be an S4 object

Creates an S4 object of the requested class.

Parameters

klass	name of the target S4 class

Exceptions

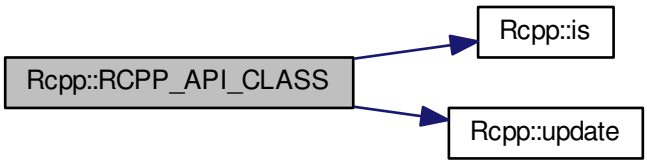
not_s4	if klass does not map to a known S4 class

Indicates if this object is an instance of the given 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:



5.2.3.227 Rcpp::RCPP_API_CLASS (Pairlist_Impl)

5.2.3.228 Rcpp::RCPP_API_CLASS (Language_Impl)

C++ wrapper around calls (LANGSXP SEXP)

This represents calls that can be evaluated

5.2.3.229 Rcpp::RCPP_API_CLASS (Environment_Impl)

5.2.3.230 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)

5.2.3.231 Rcpp::RCPP_API_CLASS (Function_Impl)

functions Finds a function. By default, searches from the global environment

Parameters

name	name of the function
env	an environment where to search the function
ns	name of the namespace in which to search the function

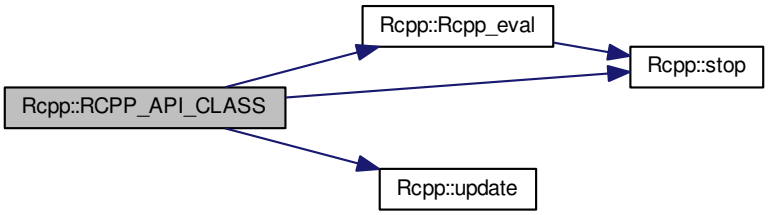
Returns the environment of this function

Returns the body of the function

Definition at line 34 of file Function.h.

References Rcpp_eval(), RCPP_GENERATE_CTOR_ASSIGN, stop(), and update().

Here is the call graph for this function:

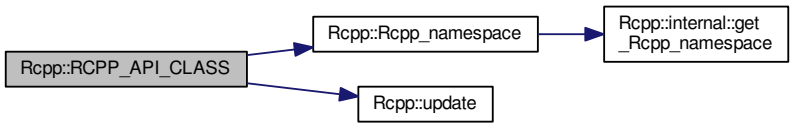


5.2.3.232 Rcpp::RCPP_API_CLASS (InternalFunction_Impl)

Definition at line 35 of file InternalFunction.h.

References RCPP_GENERATE_CTOR_ASSIGN, Rcpp_namespace(), and update().

Here is the call graph for this function:



5.2.3.233 SEXP Rcpp::Rcpp_eval (SEXP *expr,* SEXP *env*) [inline]

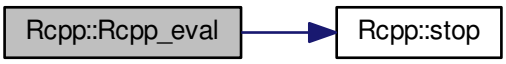
Definition at line 25 of file Rcpp_eval.h.

References stop().

Referenced by Rcpp::internal::convert_using_rfunction(), eval(), Rcpp::Vector< INTSXP >::eval(), Rcpp::DataFrame↩
_Impl< StoragePolicy >::from_list(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get(), Rcpp::FieldProxyPolicy<

CLASS >::const_FieldProxy::get(), get_last_call(), get_rcpp_cache(), init_Rcpp_cache(), 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:



5.2.3.234 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 180 of file lang.h.

References Rcpp_list9().

Here is the call graph for this function:

5.2.3.235 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 198 of file lang.h.

References Rcpp_list10().

Here is the call graph for this function:

5.2.3.236 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 216 of file lang.h.

References Rcpp_list11().

Here is the call graph for this function:

5.2.3.237 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 234 of file lang.h.

References Rcpp_list12().

Here is the call graph for this function:

5.2.3.238 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 252 of file lang.h.

References Rcpp_list13().

Here is the call graph for this function:

5.2.3.239 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 270 of file lang.h.

References Rcpp_list14().

Here is the call graph for this function:

5.2.3.240 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 288 of file lang.h.

References Rcpp_list15().

Here is the call graph for this function:

5.2.3.241 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 306 of file lang.h.

References Rcpp_list16().

Here is the call graph for this function:

5.2.3.242 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 324 of file lang.h.

References Rcpp_list17().

Here is the call graph for this function:

5.2.3.243 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 342 of file lang.h.

References Rcpp_list18().

Here is the call graph for this function:

5.2.3.244 SEXP Rcpp::Rcpp_lang2 (SEXP *x0*, SEXP *x1*) [inline]

Definition at line 40 of file lang.h.

References Rcpp_list1.

5.2.3.245 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 360 of file lang.h.

References Rcpp_list19().

Here is the call graph for this function:

5.2.3.246 SEXP Rcpp::Rcpp_lang3 (SEXP *x0*, SEXP *x1*, SEXP *x2*) [inline]

Definition at line 55 of file lang.h.

References Rcpp_list2().

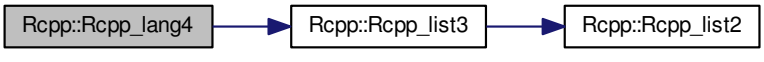
Here is the call graph for this function:

5.2.3.247 SEXP Rcpp::Rcpp_lang4 (SEXP *x0*, SEXP *x1*, SEXP *x2*, SEXP *x3*) [inline]

Definition at line 72 of file lang.h.

References Rcpp_list3().

Here is the call graph for this function:



5.2.3.248 SEXP Rcpp::Rcpp_lang5 (SEXP *x0*, SEXP *x1*, SEXP *x2*, SEXP *x3*, SEXP *x4*) [inline]

Definition at line 90 of file lang.h.

References Rcpp_list4().

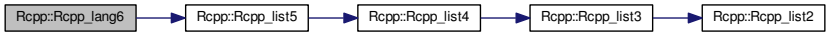
Here is the call graph for this function:

5.2.3.249 SEXP Rcpp::Rcpp_lang6 (SEXP *x0*, SEXP *x1*, SEXP *x2*, SEXP *x3*, SEXP *x4*, SEXP *x5*) [inline]

Definition at line 108 of file lang.h.

References Rcpp_list5().

Here is the call graph for this function:



5.2.3.250 SEXP Rcpp::Rcpp_lang7 (SEXP *x0*, SEXP *x1*, SEXP *x2*, SEXP *x3*, SEXP *x4*, SEXP *x5*, SEXP *x6*) [inline]

Definition at line 126 of file lang.h.

References Rcpp_list6().

Here is the call graph for this function:

5.2.3.251 SEXP Rcpp::Rcpp_lang8 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3,* SEXP *x4,* SEXP *x5,* SEXP *x6,* SEXP *x7*) [inline]

Definition at line 144 of file lang.h.

References Rcpp_list7().

Here is the call graph for this function:

5.2.3.252 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 162 of file lang.h.

References Rcpp_list8().

Here is the call graph for this function:

5.2.3.253 SEXP Rcpp::Rcpp_lcons (SEXP *car,* SEXP *cdr*) [inline]

Definition at line 30 of file lang.h.

5.2.3.254 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 173 of file lang.h.

References Rcpp_list9().

Referenced by Rcpp_lang11(), and Rcpp_list11().

Here is the call graph for this function:

5.2.3.255 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 191 of file lang.h.

References Rcpp_list10().

Referenced by Rcpp_lang12(), and Rcpp_list12().

Here is the call graph for this function:

5.2.3.256 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 209 of file lang.h.

References Rcpp_list11().

Referenced by Rcpp_lang13(), and Rcpp_list13().

Here is the call graph for this function:

5.2.3.257 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 227 of file lang.h.

References Rcpp_list12().

Referenced by Rcpp_lang14(), and Rcpp_list14().

Here is the call graph for this function:

5.2.3.258 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 245 of file lang.h.

References Rcpp_list13().

Referenced by Rcpp_lang15(), and Rcpp_list15().

Here is the call graph for this function:

5.2.3.259 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 263 of file lang.h.

References Rcpp_list14().

Referenced by Rcpp_lang16(), and Rcpp_list16().

Here is the call graph for this function:

5.2.3.260 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 281 of file lang.h.

References Rcpp_list15().

Referenced by Rcpp_lang17(), and Rcpp_list17().

Here is the call graph for this function:

5.2.3.261 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 299 of file lang.h.

References Rcpp_list16().

Referenced by Rcpp_lang18(), and Rcpp_list18().

Here is the call graph for this function:

5.2.3.262 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 317 of file lang.h.

References Rcpp_list17().

Referenced by Rcpp_lang19(), and Rcpp_list19().

Here is the call graph for this function:

5.2.3.263 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 335 of file lang.h.

References Rcpp_list18().

Referenced by Rcpp_lang20(), and Rcpp_list20().

Here is the call graph for this function:

5.2.3.264 SEXP Rcpp::Rcpp_list2 (SEXP *x0,* SEXP *x1*) [inline]

Definition at line 35 of file lang.h.

References Rcpp_list1.

Referenced by Rcpp_lang3(), and Rcpp_list3().

5.2.3.265 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 353 of file lang.h.

References Rcpp_list19().

Here is the call graph for this function:

5.2.3.266 SEXP Rcpp::Rcpp_list3 (SEXP *x0,* SEXP *x1,* SEXP *x2*) [inline]

Definition at line 49 of file lang.h.

References Rcpp_list2().

Referenced by Rcpp_lang4(), and Rcpp_list4().

Here is the call graph for this function:

5.2.3.267 SEXP Rcpp::Rcpp_list4 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3*) [inline]

Definition at line 65 of file lang.h.

References Rcpp_list3().

Referenced by Rcpp_lang5(), and Rcpp_list5().

Here is the call graph for this function:

5.2.3.268 SEXP Rcpp::Rcpp_list5 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3,* SEXP *x4*) [inline]

Definition at line 83 of file lang.h.

References Rcpp_list4().

Referenced by Rcpp_lang6(), and Rcpp_list6().

Here is the call graph for this function:

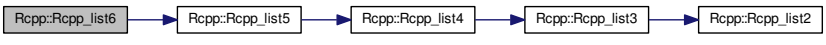
5.2.3.269 SEXP Rcpp::Rcpp_list6 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3,* SEXP *x4,* SEXP *x5*) [inline]

Definition at line 101 of file lang.h.

References Rcpp_list5().

Referenced by Rcpp_lang7(), and Rcpp_list7().

Here is the call graph for this function:



5.2.3.270 SEXP Rcpp::Rcpp_list7 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3,* SEXP *x4,* SEXP *x5,* SEXP *x6*) [inline]

Definition at line 119 of file lang.h.

References Rcpp_list6().

Referenced by Rcpp_lang8(), and Rcpp_list8().

Here is the call graph for this function:

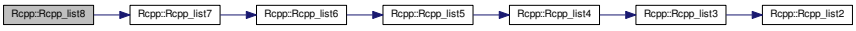
5.2.3.271 SEXP Rcpp::Rcpp_list8 (SEXP *x0,* SEXP *x1,* SEXP *x2,* SEXP *x3,* SEXP *x4,* SEXP *x5,* SEXP *x6,* SEXP *x7*) [inline]

Definition at line 137 of file lang.h.

References Rcpp_list7().

Referenced by Rcpp_lang9(), and Rcpp_list9().

Here is the call graph for this function:



5.2.3.272 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 155 of file lang.h.

References Rcpp_list8().

Referenced by Rcpp_lang10(), and Rcpp_list10().

Here is the call graph for this function:

5.2.3.273 static Environment_Impl Rcpp::Rcpp_namespace ( ) [static]

Returns

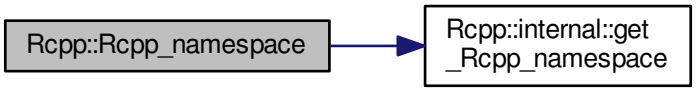
the Rcpp namespace

Definition at line 359 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:



5.2.3.274 SEXP Rcpp::Rcpp_PreserveObject (SEXP *x*) [inline]

Definition at line 84 of file RcppCommon.h.

Referenced by Rcpp_ReplaceObject(), Rcpp::String::set_encoding(), Rcpp::String::setData(), Rcpp::String::String(), and wrap< Rcpp::String >().

5.2.3.275 SEXP Rcpp::Rcpp_protect (SEXP *x*) [inline]

Definition at line 23 of file Shield.h.

5.2.3.276 void Rcpp::Rcpp_ReleaseObject (SEXP *x*) [inline]

Definition at line 91 of file RcppCommon.h.

Referenced by Rcpp_ReplaceObject(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::~PreserveStorage(), and Rcpp::String::~String().

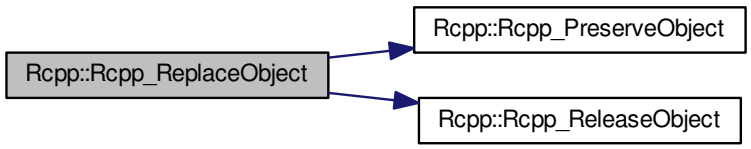
5.2.3.277 SEXP Rcpp::Rcpp_ReplaceObject (SEXP *x*, SEXP *y*) [inline]

Definition at line 97 of file RcppCommon.h.

References Rcpp_PreserveObject(), and Rcpp_ReleaseObject().

Referenced by Rcpp::String::append_wide_string(), Rcpp::String::assign_wide_string(), Rcpp::String::operator+=(), Rcpp::String::operator=(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::set__(), Rcpp::String::set_encoding(), and Rcpp::String::set_na().

Here is the call graph for this function:



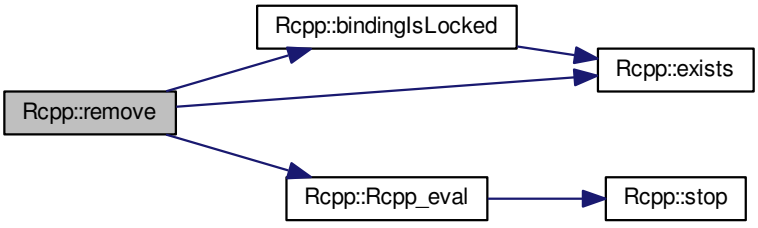
5.2.3.278 bool Rcpp::remove (const std::string & *name*)

remove an object from this environment

Definition at line 238 of file Environment.h.

References bindingIsLocked(), exists(), and Rcpp_eval().

Here is the call graph for this function:



5.2.3.279 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.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.280 sugar::Rep_Single<double> Rcpp::rep (const double & *x,* R_xlen_t *n*) [inline]

Definition at line 73 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.281 sugar::Rep_Single<int> Rcpp::rep (const int & *x,* R_xlen_t *n*) [inline]

Definition at line 76 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.282 sugar::Rep_Single<Rbyte> Rcpp::rep (const Rbyte & *x,* R_xlen_t *n*) [inline]

Definition at line 79 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.283 sugar::Rep_Single<Rcomplex> Rcpp::rep (const Rcomplex & *x,* R_xlen_t *n*) [inline]

Definition at line 82 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.284 sugar::Rep_Single<bool> Rcpp::rep (const bool & *x,* R_xlen_t *n*) [inline]

Definition at line 85 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

5.2.3.285 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.

References Rcpp::sugar::Rep_each< RTYPE, NA, T >::times.

5.2.3.286 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.

References Rcpp::sugar::Rep_len< RTYPE, NA, T >::len.

5.2.3.287 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.

5.2.3.288 NumericVector Rcpp::rexp (int *n*, double *rate*) [inline]

Definition at line 143 of file random.h.

5.2.3.289 NumericVector Rcpp::rexp (int *n*) [inline]

Definition at line 153 of file random.h.

5.2.3.290 NumericVector Rcpp::rf (int *n*, double *n1*, double *n2*) [inline]

Definition at line 157 of file random.h.

5.2.3.291 NumericVector Rcpp::rgamma (int *n*, double *a*, double *scale*) [inline]

Definition at line 171 of file random.h.

5.2.3.292 NumericVector Rcpp::rgamma (int *n*, double *a*) [inline]

Definition at line 180 of file random.h.

5.2.3.293 NumericVector Rcpp::rgeom (int *n*, double *p*) [inline]

Definition at line 189 of file random.h.

5.2.3.294 NumericVector Rcpp::rhyper (int *n*, double *nn1*, double *nn2*, double *kk*) [inline]

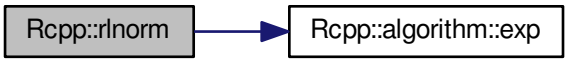
Definition at line 195 of file random.h.

5.2.3.295 NumericVector Rcpp::rlnorm (int *n*, double *meanlog*, double *sdlog*) [inline]

Definition at line 199 of file random.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:

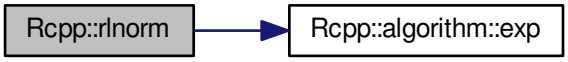


5.2.3.296 NumericVector Rcpp::rlnorm (int *n*, double *meanlog*) [inline]

Definition at line 210 of file random.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:



5.2.3.297 NumericVector Rcpp::rlnorm (int *n*) [inline]

Definition at line 221 of file random.h.

5.2.3.298 NumericVector Rcpp::rlogis (int *n*, double *location*, double *scale*) [inline]

Definition at line 225 of file random.h.

5.2.3.299 NumericVector Rcpp::rlogis (int *n,* double *location*) [inline]

Definition at line 235 of file random.h.

5.2.3.300 NumericVector Rcpp::rlogis (int *n*) [inline]

Definition at line 245 of file random.h.

5.2.3.301 NumericVector Rcpp::rnbinom (int *n,* double *siz,* double *prob*) [inline]

Definition at line 249 of file random.h.

5.2.3.302 NumericVector Rcpp::rnbinom_mu (int *n,* double *siz,* double *mu*) [inline]

Definition at line 257 of file random.h.

5.2.3.303 NumericVector Rcpp::rnchisq (int *n,* double *df,* double *lambda*) [inline]

Definition at line 264 of file random.h.

5.2.3.304 NumericVector Rcpp::rnchisq (int *n,* double *df*) [inline]

Definition at line 274 of file random.h.

5.2.3.305 NumericVector Rcpp::rnorm (int *n,* double *mean,* double *sd*) [inline]

Definition at line 64 of file random.h.

5.2.3.306 NumericVector Rcpp::rnorm (int *n,* double *mean*) [inline]

Definition at line 86 of file random.h.

5.2.3.307 NumericVector Rcpp::rnorm (int *n*) [inline]

Definition at line 102 of file random.h.

5.2.3.308 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.

5.2.3.309 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.

5.2.3.310 internal::DimNameProxy Rcpp::rownames (SEXP *x*) [inline]

Definition at line 210 of file Matrix.h.

5.2.3.311 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.

5.2.3.312 NumericVector Rcpp::rpois (int *n,* double *mu*) [inline]

Definition at line 280 of file random.h.

5.2.3.313 NumericVector Rcpp::rsignrank (int *n,* double *nn*) [inline]

Definition at line 284 of file random.h.

5.2.3.314 NumericVector Rcpp::rt (int *n,* double *df*) [inline]

Definition at line 288 of file random.h.

5.2.3.315 NumericVector Rcpp::runif (int *n,* double *min,* double *max*) [inline]

Definition at line 301 of file random.h.

Referenced by piSugar().

5.2.3.316 NumericVector Rcpp::runif (int *n,* double *min*) [inline]

Definition at line 307 of file random.h.

5.2.3.317 NumericVector Rcpp::runif (int *n*) [inline]

Definition at line 313 of file random.h.

5.2.3.318 NumericVector Rcpp::rweibull (int *n*, double *shape*, double *scale*) [inline]

Definition at line 317 of file random.h.

5.2.3.319 NumericVector Rcpp::rweibull (int *n*, double *shape*) [inline]

Definition at line 326 of file random.h.

5.2.3.320 NumericVector Rcpp::rwilcox (int *n*, double *mm*, double *nn*) [inline]

Definition at line 333 of file random.h.

5.2.3.321 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_result_of<Function, 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.

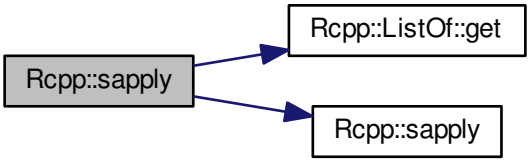
Referenced by sapply().

5.2.3.322 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:



5.2.3.323 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.

5.2.3.324 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.

5.2.3.325 Range Rcpp::seq (R_xlen_t *start,* R_xlen_t *end*) [inline]

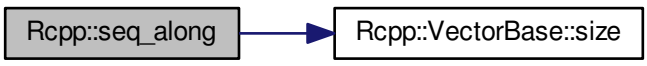
Definition at line 52 of file seq_along.h.

5.2.3.326 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:



5.2.3.327 sugar::SeqLen Rcpp::seq_len (const size_t & *n*) [inline]

Definition at line 48 of file seq_along.h.

5.2.3.328 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::VectorBase< RTYPE, na, VECTOR >::get_ref().

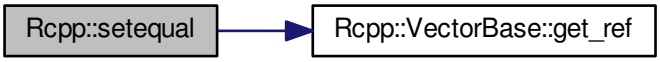
Here is the call graph for this function:

5.2.3.329 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::VectorBase< RTYPE, na, VECTOR >::get_ref().

Here is the call graph for this function:



5.2.3.330 void Rcpp::setFunction (const Function & *function*)

sets the function

Definition at line 127 of file Language.h.

5.2.3.331 void Rcpp::setSymbol (const std::string & *symbol*)

Creates a call to the given symbol using variable number of arguments

Parameters

symbol	symbol
...Args	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 111 of file Language.h.

5.2.3.332 void Rcpp::setSymbol (const Symbol & *symbol*)

sets the symbol of the call

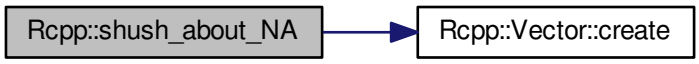
Definition at line 118 of file Language.h.

5.2.3.333 LogicalVector Rcpp::shush_about_NA ( ) [inline]

Definition at line 54 of file Na_Proxy.h.

References _, and Rcpp::Vector< RTYPE, StoragePolicy >::create().

Here is the call graph for this function:



5.2.3.334 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().

5.2.3.335 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.

5.2.3.336 template<int RTYPE, bool NA, typename T > Vector<RTYPE> Rcpp::sort_unique (const VectorBase< RTYPE, NA, T > & *t*) [inline]

Definition at line 72 of file unique.h.

5.2.3.337 template<int MAX_SIZE> std::string Rcpp::sprintf (const char * *format, ...*)

Definition at line 28 of file sprintf.h.

5.2.3.338 template<typename T > void Rcpp::standard_delete_finalizer (T * *obj*)

Definition at line 30 of file XPtr.h.

5.2.3.339 void Rcpp::stop (const std::string & *message*) [inline]

Definition at line 267 of file exceptions.h.

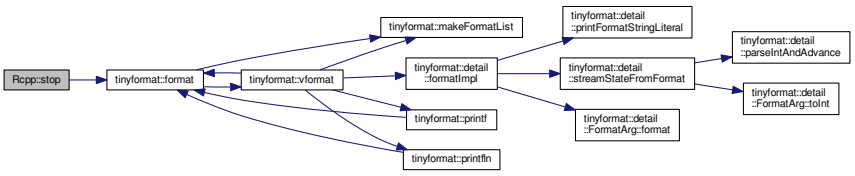
Referenced by Rcpp::internal::DimNameProxy::assign(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices(), Rcpp::Vector< INTSXP >::findName(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), new_env(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=(), RCPP_API_CLASS(), and Rcpp_eval().

5.2.3.340 template<typename T1 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1*) [inline]

Definition at line 272 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

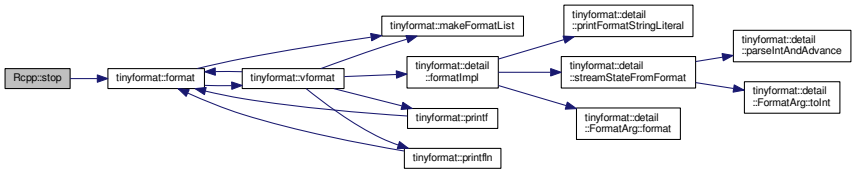


5.2.3.341 template<typename T1 , typename T2 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2*) [inline]

Definition at line 277 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

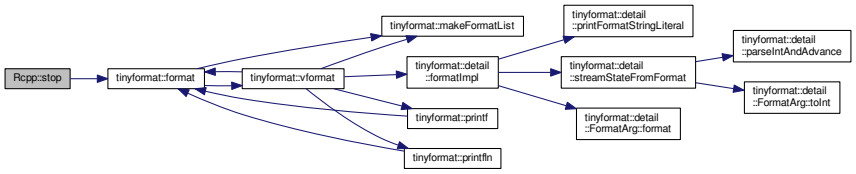


5.2.3.342 template<typename T1 , typename T2 , typename T3 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3*) [inline]

Definition at line 282 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

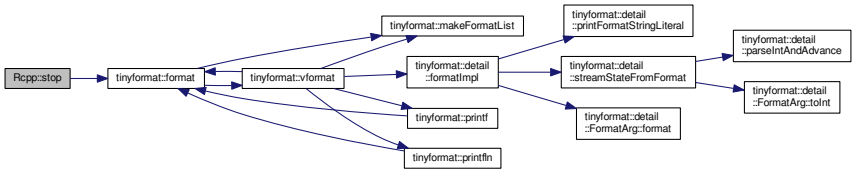


5.2.3.343 template<typename T1 , typename T2 , typename T3 , typename T4 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4*) [inline]

Definition at line 287 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

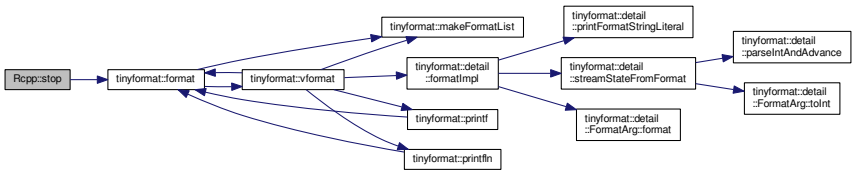


5.2.3.344 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5*) [inline]

Definition at line 292 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

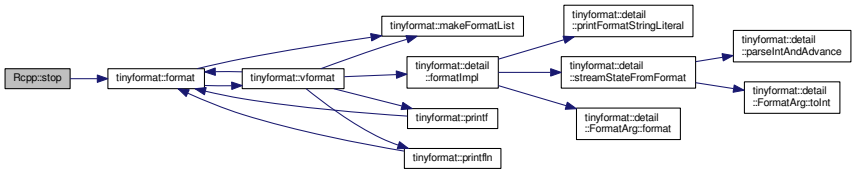


5.2.3.345 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6*) [inline]

Definition at line 297 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

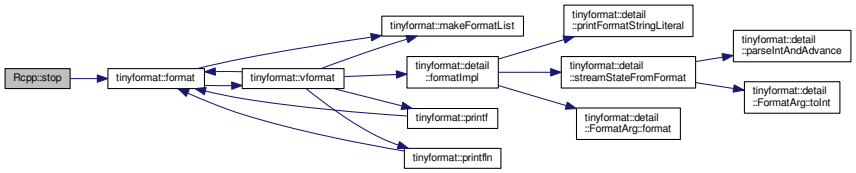


5.2.3.346 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7*) [inline]

Definition at line 302 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

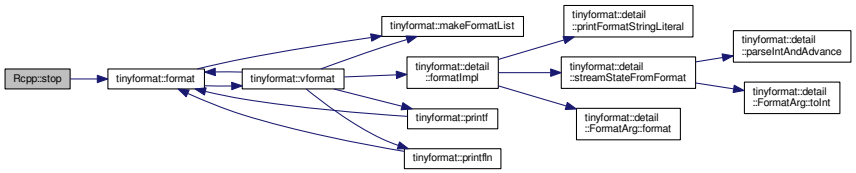


5.2.3.347 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8*) [inline]

Definition at line 307 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

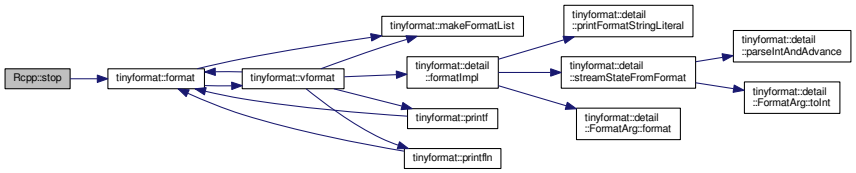


5.2.3.348 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8,* const T9 & *arg9*) [inline]

Definition at line 312 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

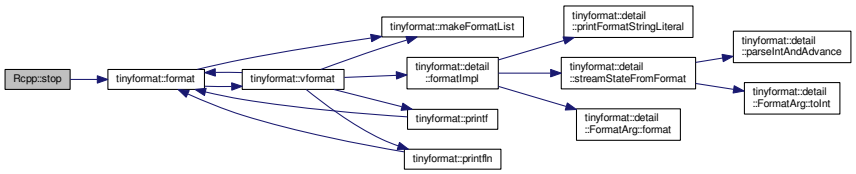


5.2.3.349 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 > void NORET Rcpp::stop (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8,* const T9 & *arg9,* const T10 & *arg10*) `[inline]`

Definition at line 317 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:



5.2.3.350 Rcpp::StretchyList_Impl ( )

Definition at line 42 of file StretchyList.h.

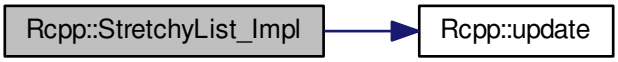
Referenced by push_front().

5.2.3.351 Rcpp::StretchyList_Impl (SEXP *x*)

Definition at line 47 of file StretchyList.h.

References update().

Here is the call graph for this function:



5.2.3.352 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(), and piSugar().

5.2.3.353 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.

5.2.3.354 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.

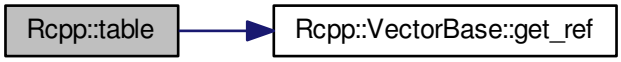
5.2.3.355 template<int RTYPE, bool NA, typename T > IntegerVector Rcpp::table (const VectorBase< RTYPE, NA, T > & *x*) [inline]

Definition at line 126 of file table.h.

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

Referenced by in(), and match().

Here is the call graph for this function:



5.2.3.356 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.

References Rcpp::sugar::Tail< RTYPE, NA, T >::n.

Referenced by Rcpp::DottedPairImpl< CLASS >::insert(), and Rcpp::DottedPairImpl< CLASS >::push_back().

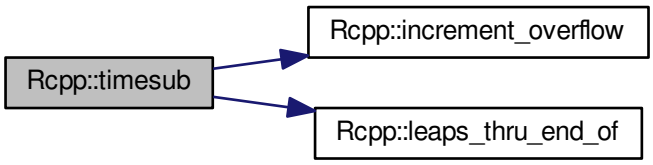
5.2.3.357 static struct tm * Rcpp::timesub (const time_t *const *timep*, const long *offset*, const struct state *const *sp*, struct tm *const *tmp*) [static]

Definition at line 1135 of file Date.cpp.

References DAYSPERLYEAR, DAYSPERNYEAR, DAYSPERWEEK, 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, SECSPERDAY, SECSPERHOUR, and SECSPERMIN.

Referenced by gmtsub().

Here is the call graph for this function:



5.2.3.358 static std::string Rcpp::toString (const int *i*) [static]

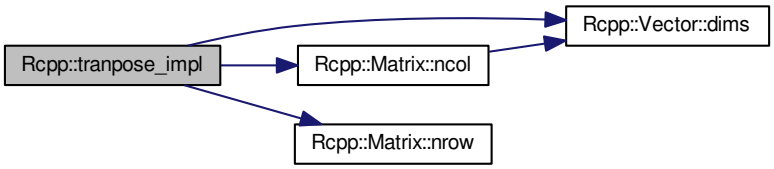
Definition at line 44 of file exceptions.h.

5.2.3.359 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.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:



5.2.3.360 template<template< class > class StoragePolicy> Matrix<REALSXP, StoragePolicy> Rcpp::transpose (const Matrix< REALSXP, StoragePolicy > & *x*)

Definition at line 435 of file Matrix.h.

5.2.3.361 template<template< class > class StoragePolicy> Matrix<INTSXP, StoragePolicy> Rcpp::transpose (const Matrix< INTSXP, StoragePolicy > & *x*)

Definition at line 440 of file Matrix.h.

5.2.3.362 template<template< class > class StoragePolicy> Matrix<STRSXP, StoragePolicy> Rcpp::transpose (const Matrix< STRSXP, StoragePolicy > & *x*)

Definition at line 445 of file Matrix.h.

5.2.3.363 static time_t Rcpp::transtime (const time_t *janfirst,* const int *year,* const struct rule *const *rulep,* const long *offset*) [static]

Definition at line 796 of file Date.cpp.

References DAY_OF_YEAR, DAYSPERWEEK, INITIALIZE, isleap, JULIAN_DAY, 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().

5.2.3.364 const char * Rcpp::type2name (SEXP *x*) [inline]

Definition at line 70 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.2.3.365 static int Rcpp::typesequiv (const struct state * *sp,* int *a,* int *b*) [static]

Definition at line 1110 of file Date.cpp.

References Rcpp::state::chars, Rcpp::ttinfo::tt_abbrind, Rcpp::ttinfo::tt_gmtoff, Rcpp::ttinfo::tt_isdst, Rcpp::ttinfo::tt_↩ ttisgmt, Rcpp::ttinfo::tt_ttisstd, Rcpp::state::ttis, and Rcpp::state::typecnt.

Referenced by tzload().

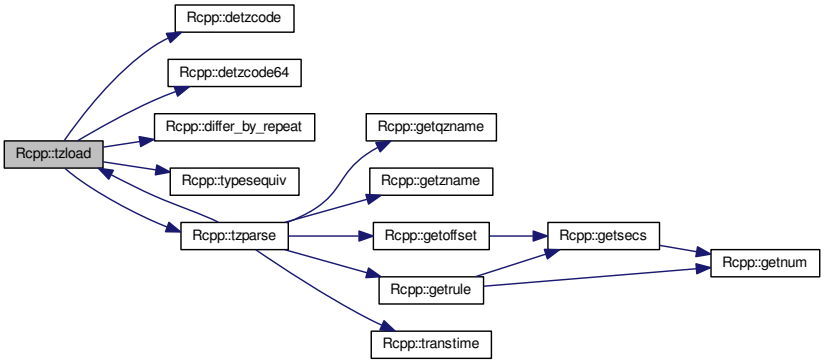
5.2.3.366 static int Rcpp::tzload (const char * *name,* struct state *const *sp,* const int *doextend*) [static]

Definition at line 553 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::lsinfo::ls_corr, Rcpp::lsinfo::ls_trans, Rcpp↩ ::state::lsis, OPEN_MODE, Rcpp::state::timecnt, Rcpp::ttinfo::tt_abbrind, Rcpp::ttinfo::tt_gmtoff, Rcpp::ttinfo::tt_↩ isdst, Rcpp::ttinfo::tt_ttisgmt, Rcpp::ttinfo::tt_ttisstd, 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 YEARSPERREPEAT.

Referenced by gmtload(), and tzparse().

Here is the call graph for this function:



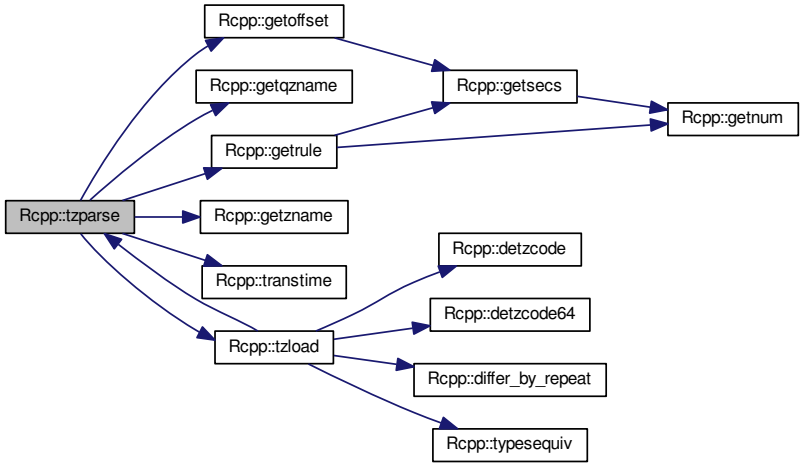
5.2.3.367 static int Rcpp::tzparse (const char * *name,* struct **state** * *sp,* int *lastditch*) [static]

Definition at line 880 of file Date.cpp.

References Rcpp::state::ats, Rcpp::state::charcnt, Rcpp::state::chars, EPOCH_YEAR, getoffset(), getqzname(), getrule(), getzname(), INITIALIZE, isleap, Rcpp::state::leapcnt, SECSPERDAY, SECSPERHOUR, Rcpp::state::timecnt, transtime(), Rcpp::ttinfo::tt_abbrind, Rcpp::ttinfo::tt_gmtoff, Rcpp::ttinfo::tt_isdst, Rcpp::ttinfo::tt_ttisgmt, Rcpp::ttinfo::tt_ttisstd, Rcpp::state::ttis, Rcpp::state::typecnt, Rcpp::state::types, TZ_MAX_TIMES, TZDEFRULES, TZDEFRULESTRING, and tzload().

Referenced by gmtload(), and tzload().

Here is the call graph for this function:

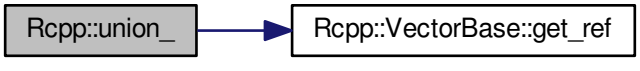


5.2.3.368 template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T > **Vector**<RTYPE> **Rcpp::union_** (const **VectorBase**< RTYPE, LHS_NA, LHS_T > & *lhs,* const **VectorBase**< RTYPE, RHS_NA, RHS_T > & *rhs*) [inline]

Definition at line 177 of file setdiff.h.

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

Here is the call graph for this function:

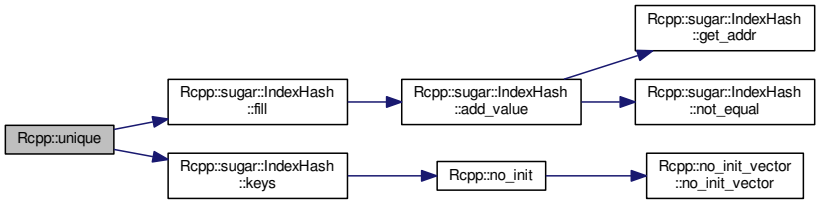


5.2.3.369 template<int RTYPE, bool NA, typename T > Vector<RTYPE> Rcpp::unique (const VectorBase< RTYPE, NA, T > & *t*) [inline]

Definition at line 65 of file unique.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::InSet< HASH >::hash, and Rcpp::sugar::IndexHash< RTYPE >::keys().

Here is the call graph for this function:



5.2.3.370 void Rcpp::unlockBinding (const std::string & *name*)

unlocks the given binding see ?bindingIsLocked

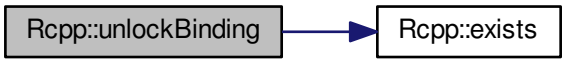
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 285 of file Environment.h.

References exists().

Here is the call graph for this function:



5.2.3.371 void Rcpp::update (SEXP *x*)

Definition at line 398 of file Environment.h.

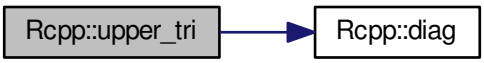
Referenced by fast_eval(), Formula_Impl(), Pairlist_Impl(), RCPP_API_CLASS(), and StretchyList_Impl().

5.2.3.372 template<int RTYPE, bool LHS_NA, typename LHS_T > sugar::UpperTri<RTYPE,LHS_NA,LHS_T> Rcpp::upper_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > & *lhs,* bool *diag* = `false`) [inline]

Definition at line 70 of file upper_tri.h.

References diag().

Here is the call graph for this function:



5.2.3.373 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().

5.2.3.374 template<bool NA, typename T > sugar::Var<INTSXP,NA,T> Rcpp::var (const VectorBase< INTSXP, NA, T > & *t*) [inline]

Definition at line 79 of file var.h.

5.2.3.375 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.

5.2.3.376 template<bool NA, typename T > sugar::Var<CPLXSXP,NA,T> Rcpp::var (const VectorBase< CPLXSXP, NA, T > & *t*) [inline]

Definition at line 89 of file var.h.

5.2.3.377 void Rcpp::warning (const std::string & *message*) [inline]

Definition at line 213 of file exceptions.h.

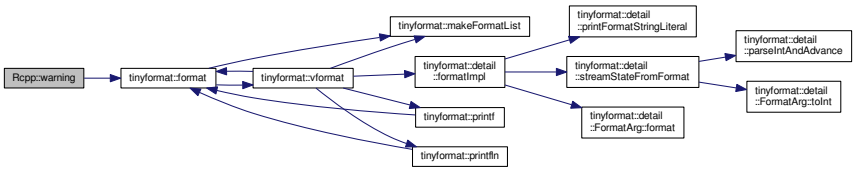
Referenced by Rcpp::attributes::showWarning().

5.2.3.378 template<typename T1 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1*) [inline]

Definition at line 218 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

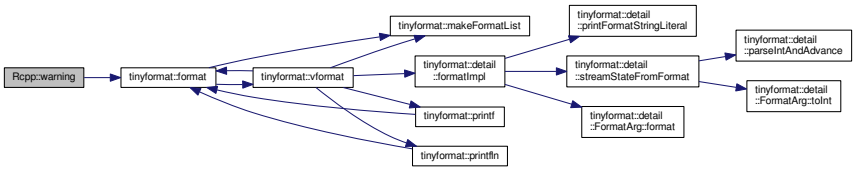


5.2.3.379 template<typename T1 , typename T2 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2*) [inline]

Definition at line 223 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

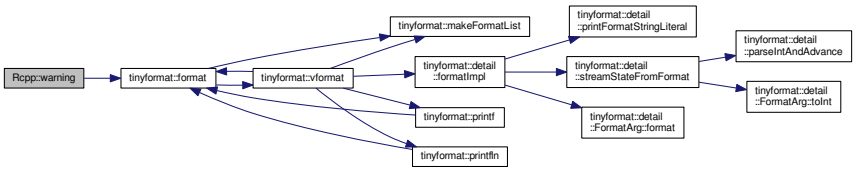


5.2.3.380 template<typename T1 , typename T2 , typename T3 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3*) [inline]

Definition at line 228 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

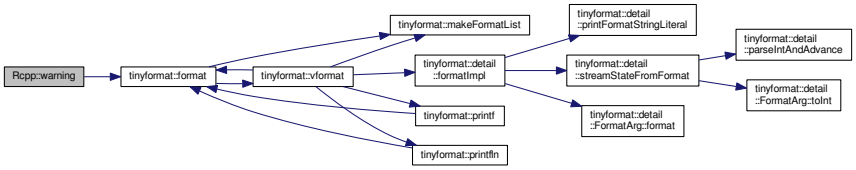


5.2.3.381 template<typename T1 , typename T2 , typename T3 , typename T4 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4*) [inline]

Definition at line 233 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

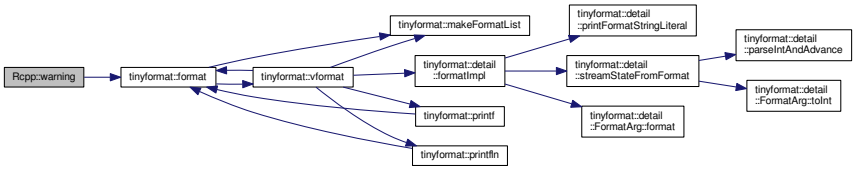


5.2.3.382 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5*) [inline]

Definition at line 238 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

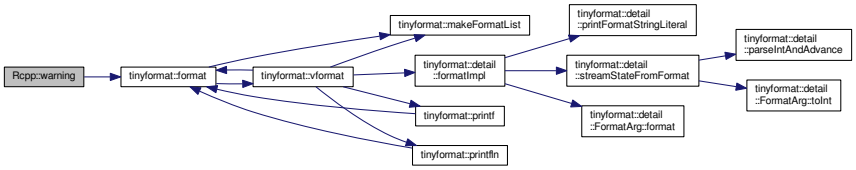


5.2.3.383 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6*) [inline]

Definition at line 243 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

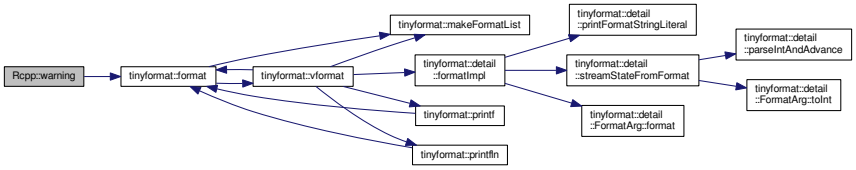


5.2.3.384 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7*) [inline]

Definition at line 248 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

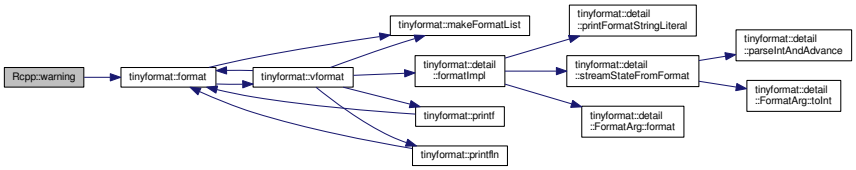


5.2.3.385 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8*) [inline]

Definition at line 253 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

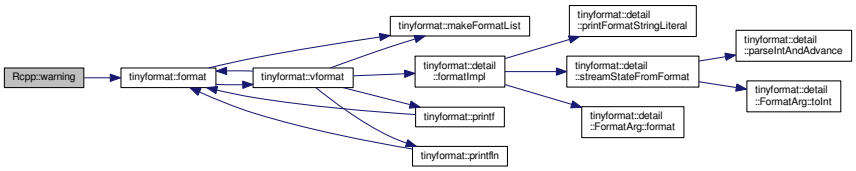


5.2.3.386 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8,* const T9 & *arg9*) [inline]

Definition at line 258 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:

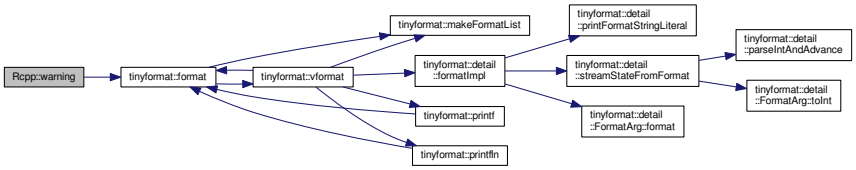


5.2.3.387 template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 > void Rcpp::warning (const char * *fmt,* const T1 & *arg1,* const T2 & *arg2,* const T3 & *arg3,* const T4 & *arg4,* const T5 & *arg5,* const T6 & *arg6,* const T7 & *arg7,* const T8 & *arg8,* const T9 & *arg9,* const T10 & *arg10*) [inline]

Definition at line 263 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:



5.2.3.388 void Rcpp::warningcall (SEXP *call,* const std::string & *s*) [inline]

Definition at line 29 of file print.h.

5.2.3.389 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.

5.2.3.390 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.

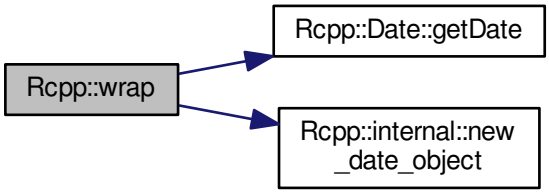
5.2.3.391 template<> SEXP Rcpp::wrap (const Date & *date*) [inline]

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< INTSXP >::assign(), Rcpp::Vector< INTSXP >::assign_object(), Rcpp::Vector< INTSXP >::assign_sugar_expression(), Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl(), Environment_Impl(), Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), CppProperty_Getter_Setter< PROP >::get(), CppProperty_GetMethod< Class, PROP >::get(), CppProperty_Getter< PROP >::get(), CppProperty_GetConstMethod< Class, PROP >::get(), CppProperty_GetPointerMethod< Class, PROP >::get(), Rcpp::internal::generic_element_converter< RTYPE >::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::sugar::IndexHash< RTYPE >::get_profile_data(), Rcpp::internal::grow__dispatch(), Rcpp::Dimension::operator SEXP(), Rcpp::internal::simple_name_proxy< RTYPE >::operator SEXP(), Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP(), Rcpp::ListOf< T >::operator SEXP(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator SEXP(), Rcpp::Armor< T >::operator=(), Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=(), Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator=(), Rcpp::internal::generic_proxy< RTYPE >::operator=(), Rcpp::ChildVector< T >::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::internal::generic_name_proxy< RTYPE >::operator=(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=(), Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[](), PiLeibniz(), Rcpp::internal::range_wrap_dispatch___impl__pair(), RCPP_API_CLASS(), and Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser().

Here is the call graph for this function:

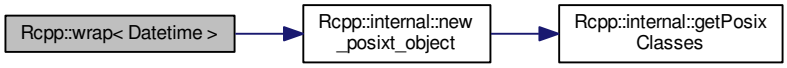


5.2.3.392 template<> SEXP Rcpp::wrap< Datetime > (const Datetime & *date*) [inline]

Definition at line 44 of file Datetime.h.

References Rcpp::internal::new_posixt_object().

Here is the call graph for this function:



5.2.3.393 template<> SEXP Rcpp::wrap< Rcpp::Date > (const Rcpp::Date & *date*)

5.2.3.394 template<> SEXP Rcpp::wrap< Rcpp::Datetime > (const Rcpp::Datetime & *dt*)

5.2.3.395 template<> SEXP Rcpp::wrap< Rcpp::String > (const Rcpp::String & *object*) [inline]

Definition at line 511 of file String.h.

References Rcpp::String::data, Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Here is the call graph for this function:

5.2.3.396 template<typename T > SEXP Rcpp::wrap_extra_steps (SEXP *x*)

Definition at line 28 of file wrap_extra_steps.h.

5.2.3.397 template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Date > (SEXP *x*) [inline]

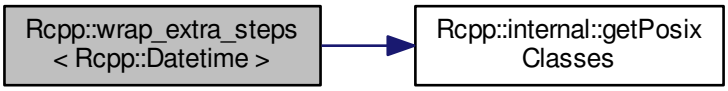
Definition at line 131 of file Date.h.

5.2.3.398 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::Module::invoke(), shush_about_NA(), sourceCppContext(), and stack_trace().

5.2.4.2 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> res( Rcpp_eval( Rf_lang2( asEnvironmentSym, x ) ) );
                return res ;
            } catch( const eval_error& ex){
                throw not_compatible( "cannot convert to environment"  ) ;
            }
        }

    public:


        Environment_Impl(){
            Storage::set__(R_GlobalEnv) ;
        }
```

Definition at line 34 of file Environment.h.

5.2.4.3 Module* Rcpp::current_scope [static]

Definition at line 224 of file Module.cpp.

Referenced by getCurrentScope(), and setCurrentScope().

5.2.4.4 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 public public Rcpp::DottedPairImpl< Formula_Impl< StoragePolicy > >

Initial value:

```
{
    public:


        Formula_Impl(){}
```

Definition at line 33 of file Formula.h.

5.2.4.6 public public Rcpp::DottedPairImpl< Language_Impl< StoragePolicy > >

Initial value:

```
{
    public:

        typedef typename DottedPairProxyPolicy<Language_Impl>::DottedPairProxy Proxy
```

Definition at line 35 of file Language.h.

5.2.4.7 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 public Rcpp::DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >

Definition at line 28 of file DottedPair.h.

5.2.4.9 public Rcpp::DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >

Definition at line 31 of file Formula.h.

5.2.4.10 public Rcpp::DottedPairProxyPolicy< Language_Impl< StoragePolicy > >

Definition at line 33 of file Language.h.

5.2.4.11 public Rcpp::DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >

Definition at line 32 of file Pairlist.h.

5.2.4.12 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 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> call( Rf_lang2( newSym, Rf_mkString( klass.c_str() ) ) );
            Storage::set__( Rcpp_eval( call ,
      Rcpp::internal::get_Rcpp_namespace()) );
        }

        void update( SEXP x){
            if( ! ::Rf_isS4(x) ) throw not_reference();
        }
    }
```

Definition at line 32 of file Reference.h.

5.2.4.14 const char Rcpp::gmt[ ] = "GMT" [static]

Definition at line 299 of file Date.cpp.

5.2.4.15 int Rcpp::gmt_is_set [static]

Definition at line 370 of file Date.cpp.

5.2.4.16 struct state Rcpp::gmtmem [static]

Definition at line 373 of file Date.cpp.

5.2.4.17 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 361 of file Date.cpp.

5.2.4.18 Na_Proxy Rcpp::NA [static]

Definition at line 52 of file Na_Proxy.h.

Referenced by Rcpp::Vector< INTSXP >::import_sugar_expression(), Rcpp::algorithm::helpers::rtype< T >::NA(), operator!=(), operator<(), operator<=(), operator==(), operator>(), operator>=(), and sapply().

5.2.4.19 Rostream<false> Rcpp::Rcerr [static]

Definition at line 87 of file Rstreambuf.h.

5.2.4.20 Rostream<true> Rcpp::Rcout [static]

Definition at line 86 of file Rstreambuf.h.

Referenced by compileAttributes(), and Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

5.2.4.21 struct tm Rcpp::tm [static]

Definition at line 377 of file Date.cpp.

Referenced by gmtime_().

5.2.4.22 const int Rcpp::year_lengths[2] [static]

Initial value:

```
= {
        DAYSPERNYEAR, DAYSPERLYEAR
    }
```

Definition at line 366 of file Date.cpp.

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 template<typename InputIterator , typename OutputIterator > void Rcpp::algorithm::exp (InputIterator *begin,* InputIterator *end,* OutputIterator *out*)

Definition at line 455 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 template<typename InputIterator , typename OutputIterator > void Rcpp::algorithm::log (InputIterator *begin,* InputIterator *end,* OutputIterator *out*)

Definition at line 450 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::WeibullGenerator::operator()(), Rcpp::stats::WeibullGenerator__scale1::operator()(), Rcpp::stats::LogisGenerator_1::operator()(), Rcpp::stats::LogisGenerator_0::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().

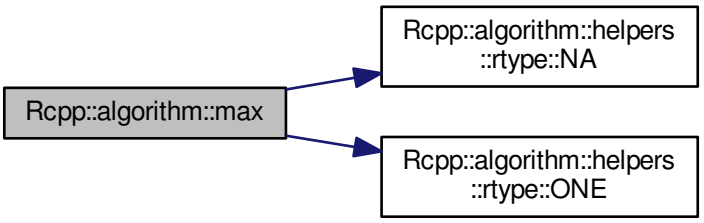
5.3.1.3 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::max (InputIterator *begin,* InputIterator *end*)

Definition at line 305 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA(), Rcpp::algorithm::helpers::rtype< T >::ONE(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Referenced by max_nona(), and Rcpp::operator<<().

Here is the call graph for this function:



5.3.1.4 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 330 of file algorithm.h.

References max(), Rcpp::algorithm::helpers::rtype< T >::ONE(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

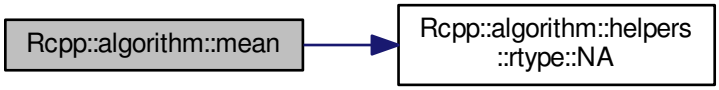
Here is the call graph for this function:

5.3.1.5 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 398 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Here is the call graph for this function:

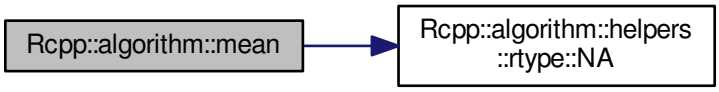


5.3.1.6 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value && traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type, int >::value, double >::type Rcpp::algorithm::mean (InputIterator *begin,* InputIterator *end*)

Definition at line 425 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA().

Here is the call graph for this function:



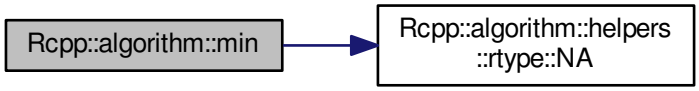
5.3.1.7 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::min (InputIterator *begin,* InputIterator *end*)

Definition at line 351 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Referenced by tinyformat::detail::formatTruncated(), min_nona(), Rcpp::operator<<(), and PiLeibniz().

Here is the call graph for this function:

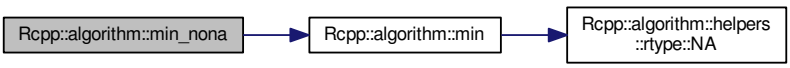


5.3.1.8 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::min_nona (InputIterator *begin,* InputIterator *end*)

Definition at line 376 of file algorithm.h.

References min(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Here is the call graph for this function:

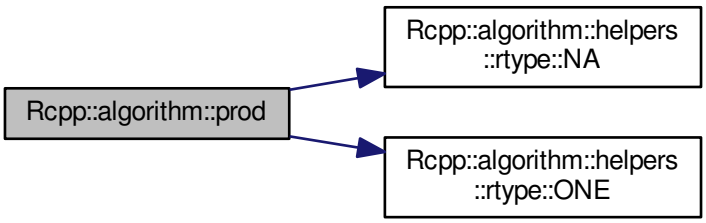


5.3.1.9 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::prod (InputIterator *begin,* InputIterator *end*)

Definition at line 259 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA(), Rcpp::algorithm::helpers::rtype< T >::ONE(), and Rcpp↩ ::algorithm::helpers::ctype_helper< I >::value.

Here is the call graph for this function:

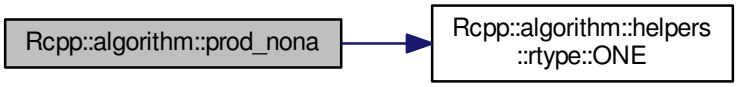


5.3.1.10 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::prod_nona (InputIterator *begin,* InputIterator *end*)

Definition at line 284 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::ONE(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Here is the call graph for this function:



5.3.1.11 template<typename InputIterator , typename OutputIterator > void Rcpp::algorithm::sqrt (InputIterator *begin,* InputIterator *end,* OutputIterator *out*)

Definition at line 460 of file algorithm.h.

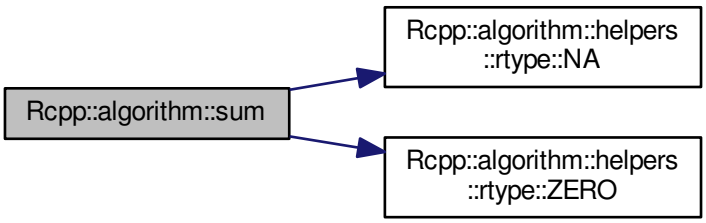
Referenced by Rcpp::internal::complex__asin(), Rcpp::internal::complex__Mod(), Rcpp::internal::complex__sqrt(), fastLm(), Rcpp::sugar::Sd< RTYPE, NA, T >::get(), Rcpp::stats::TGenerator::operator()(), Rcpp::algorithm::helpers↩ ::sqrt::operator()(), and piSugar().

5.3.1.12 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 213 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA(), Rcpp::algorithm::helpers::ctype_helper< I >::value, and Rcpp::algorithm::helpers::rtype< T >::ZERO().

Here is the call graph for this function:

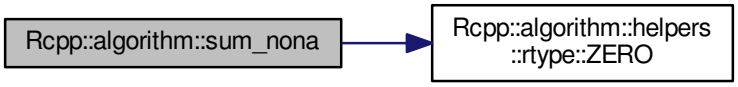


5.3.1.13 template<typename InputIterator > traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::sum_nona (InputIterator *begin,* InputIterator *end*)

Definition at line 238 of file algorithm.h.

References Rcpp::algorithm::helpers::ctype_helper< I >::value, and Rcpp::algorithm::helpers::rtype< T >::ZERO().

Here is the call graph for this function:



5.4 Rcpp::algorithm::helpers Namespace Reference

Classes

- struct ctype
- struct CTYPE_CHAR
- struct CTYPE_DOUBLE
- struct CTYPE_FLOAT
- struct ctype_helper
- struct ctype_helper< sizeof(CTYPE_CHAR) >
- struct ctype_helper< sizeof(CTYPE_DOUBLE) >
- struct ctype_helper< sizeof(CTYPE_FLOAT) >
- struct ctype_helper< sizeof(CTYPE_INT) >
- struct ctype_helper< sizeof(CTYPE_LONG) >
- struct ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >
- struct ctype_helper< sizeof(CTYPE_SHORT) >
- struct ctype_helper< sizeof(CTYPE_STRING) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >
- struct CTYPE_INT
- struct CTYPE_LONG
- struct CTYPE_LONG_DOUBLE
- struct CTYPE_SHORT
- struct CTYPE_STRING
- struct CTYPE_UNKNOWN
- struct CTYPE_UNSIGNED_CHAR

- struct CTYPE_UNSIGNED_INT
- struct CTYPE_UNSIGNED_LONG
- struct CTYPE_UNSIGNED_SHORT
- struct decays_to_ctype
- struct exp
- struct log
- struct rtype
- struct rtype_helper
- struct rtype_helper< double >
- struct rtype_helper< int >
- struct sqrt

5.5 Rcpp::attributes Namespace Reference

Classes

- class Argument
- class Attribute
- class CommentState
- class CppExportsGenerator
- class CppExportsIncludeGenerator
- class CppPackageIncludeGenerator
- class ExportsGenerator
- class ExportsGenerators
- class FileInfo
- class Function
- class Param
- class RExportsGenerator
- class SourceFileAttributes
- class SourceFileAttributesParser
- class Type

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)
- 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)
- 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 = " \f\n\r\t\v"
- const char *const kExportAttribute = "export"
- const char *const kExportName = "name"
- const char *const kExportRng = "rng"
- 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"

5.5.1 Function Documentation

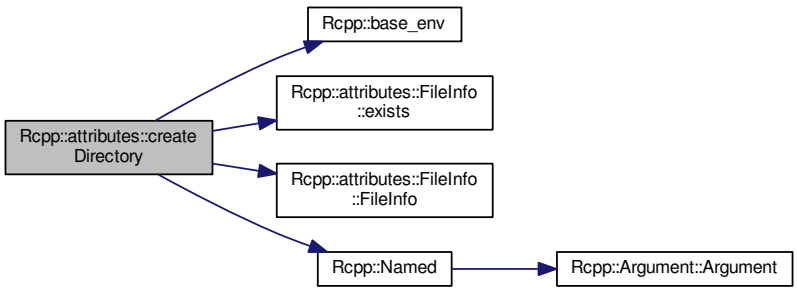
5.5.1.1 void Rcpp::attributes::createDirectory (const std::string & *path*)

Definition at line 2707 of file attributes.cpp.

References Rcpp::base_env(), Rcpp::attributes::FileInfo::exists(), Rcpp::attributes::FileInfo::FileInfo(), and Rcpp::Named().

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), and Rcpp::attributes::CppPackageIncludeGenerator::commit().

Here is the call graph for this function:



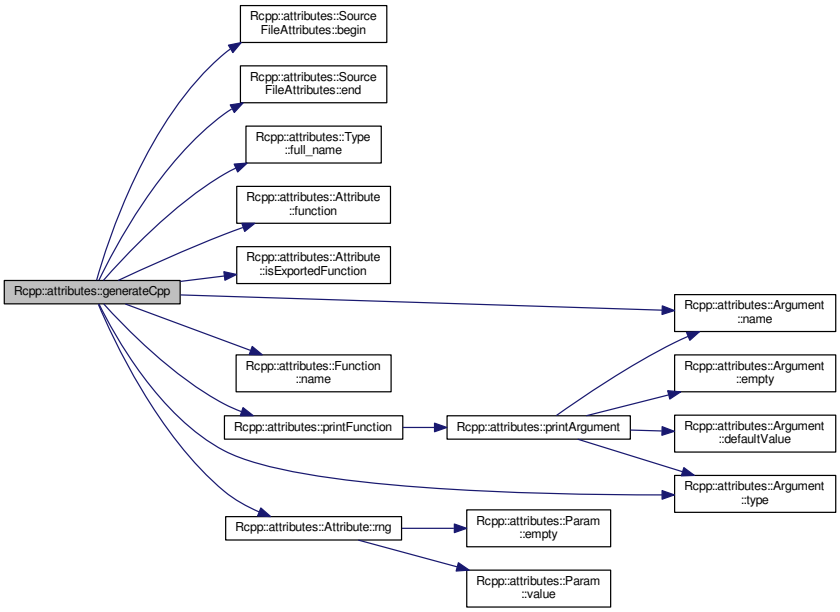
5.5.1.2 void Rcpp::attributes::generateCpp (std::ostream & *ostr,* const SourceFileAttributes & *attributes,* bool *includePrototype,* bool *cppInterface,* const std::string & *contextId*)

Definition at line 2541 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(), and isRoxygenCpp().

Here is the call graph for this function:



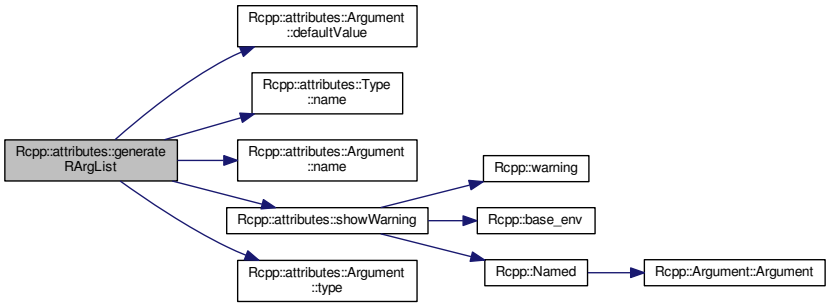
5.5.1.3 std::string Rcpp::attributes::generateRArgList (const Function & *function*)

Definition at line 2514 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::RExportsGenerator::doWriteFunctions(), and isRoxygenCpp().

Here is the call graph for this function:



5.5.1.4 bool Rcpp::attributes::isQuoted (const std::string & *str*)

Definition at line 2794 of file attributes.cpp.

Referenced by Rcpp::attributes::ExportsGenerators::remove().

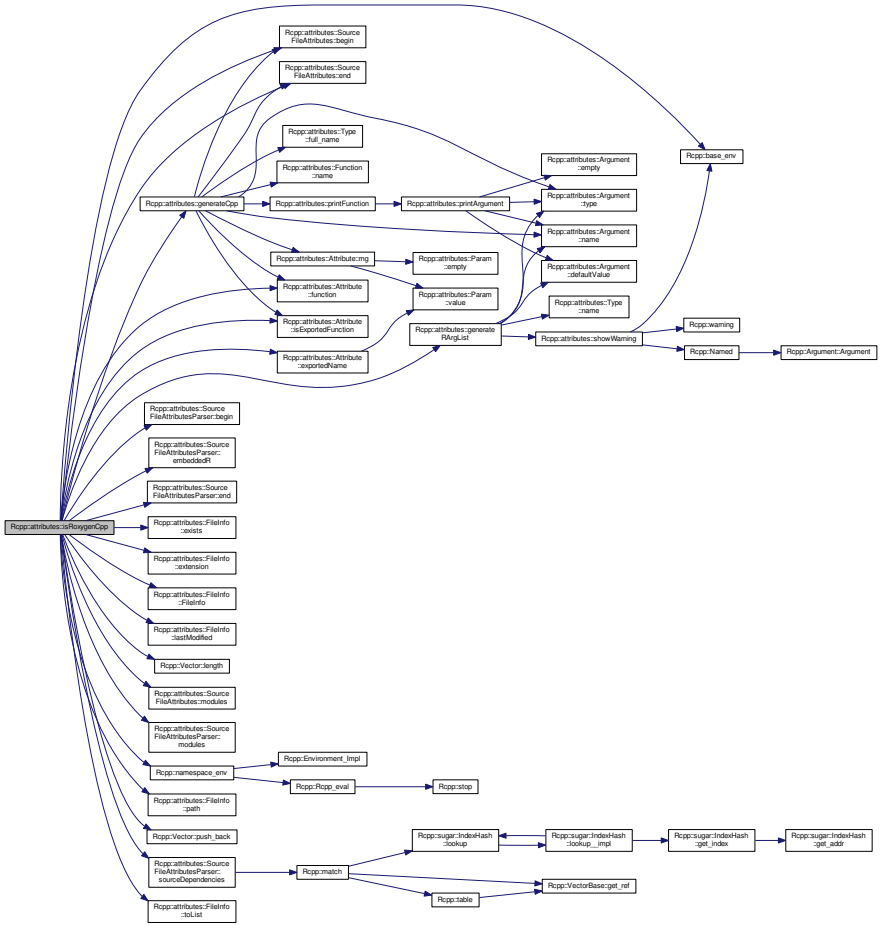
5.5.1.5 bool Rcpp::attributes::isRoxygenCpp (const std::string & *str*)

Definition at line 2807 of file attributes.cpp.

References Rcpp::base_env(), Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes↩
Parser::begin(), Rcpp::attributes::SourceFileAttributesParser::embeddedR(), Rcpp::attributes::SourceFileAttributes↩
::end(), Rcpp::attributes::SourceFileAttributesParser::end(), Rcpp::attributes::FileInfo::exists(), Rcpp::attributes↩
::Attribute::exportedName(), Rcpp::attributes::FileInfo::extension(), Rcpp::attributes::FileInfo::FileInfo(), Rcpp↩
::attributes::Attribute::function(), generateCpp(), generateRArgList(), Rcpp::attributes::Attribute::isExportedFunction(), kDependsAttribute, kExportAttribute, kPluginsAttribute, Rcpp::attributes::FileInfo::lastModified(), Rcpp::Vector< R↩
TYPE, StoragePolicy >::length(), Rcpp::attributes::SourceFileAttributes::modules(), Rcpp::attributes::SourceFile↩
AttributesParser::modules(), Rcpp::namespace_env(), Rcpp::attributes::FileInfo::path(), Rcpp::Vector< RTYP↩
E, StoragePolicy >::push_back(), Rcpp::attributes::SourceFileAttributesParser::sourceDependencies(), and Rcpp↩
::attributes::FileInfo::toList().

Referenced by stripTrailingLineComments().

Here is the call graph for this function:



5.5.1.6 bool Rcpp::attributes::isWhitespace (char *ch*)

Definition at line 2718 of file attributes.cpp.

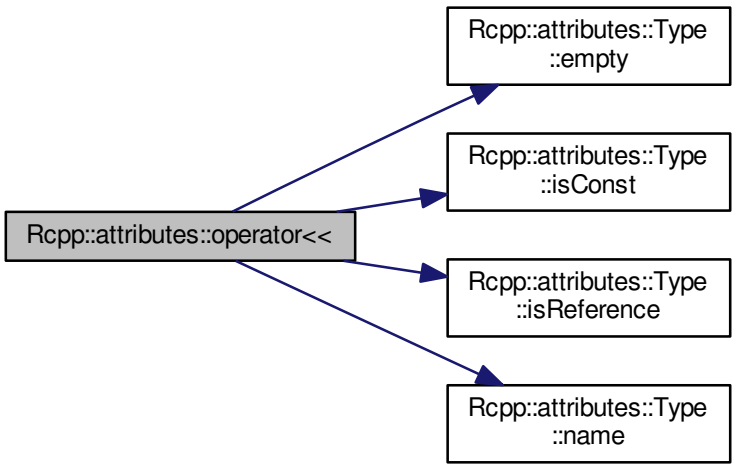
Referenced by Rcpp::attributes::SourceFileAttributesParser::parseFunction().

5.5.1.7 std::ostream & Rcpp::attributes::operator<< (std::ostream & *os,* const Type & *type*)

Definition at line 1024 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:

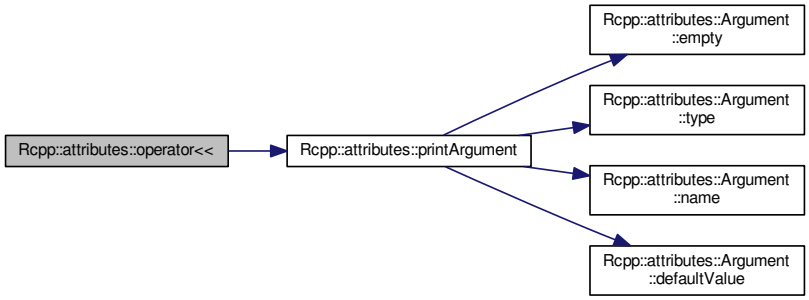


5.5.1.8 std::ostream & Rcpp::attributes::operator<< (std::ostream & *os*, const Argument & *argument*)

Definition at line 1051 of file attributes.cpp.

References printArgument().

Here is the call graph for this function:

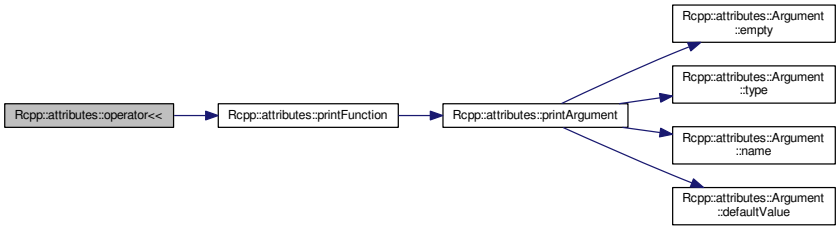


5.5.1.9 std::ostream & Rcpp::attributes::operator<< (std::ostream & *os,* const Function & *function*)

Definition at line 1079 of file attributes.cpp.

References printFunction().

Here is the call graph for this function:

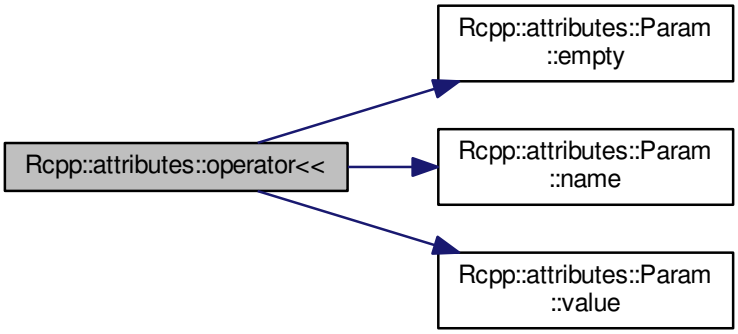


5.5.1.10 std::ostream & Rcpp::attributes::operator<< (std::ostream & *os,* const Param & *param*)

Definition at line 1085 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:

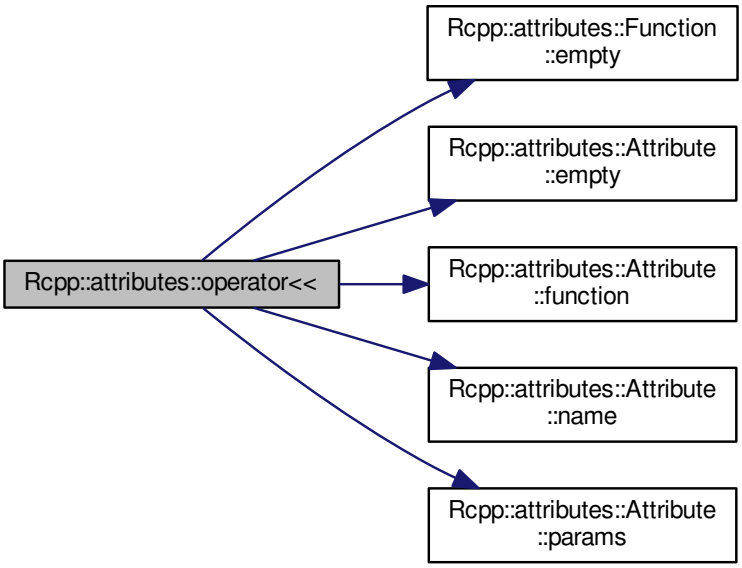


5.5.1.11 std::ostream & Rcpp::attributes::operator<< (std::ostream & *os*, const **Attribute** & *attribute*)

Definition at line 1095 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:



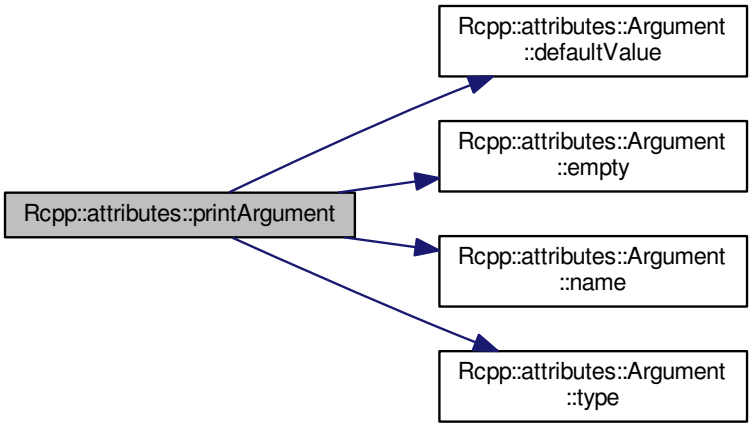
5.5.1.12 void Rcpp::attributes::printArgument (std::ostream & *os*, const **Argument** & *argument*, bool *printDefault* = `true`)

Definition at line 1036 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:



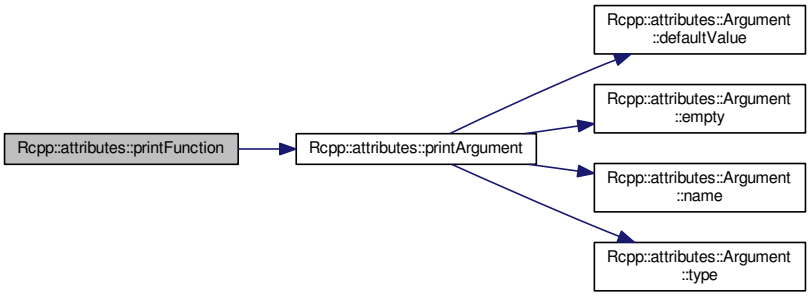
5.5.1.13 void Rcpp::attributes::printFunction (std::ostream & *os,* const Function & *function,* bool *printArgDefaults* = `true`)

Definition at line 1057 of file attributes.cpp.

References printArgument().

Referenced by generateCpp(), and operator<<().

Here is the call graph for this function:



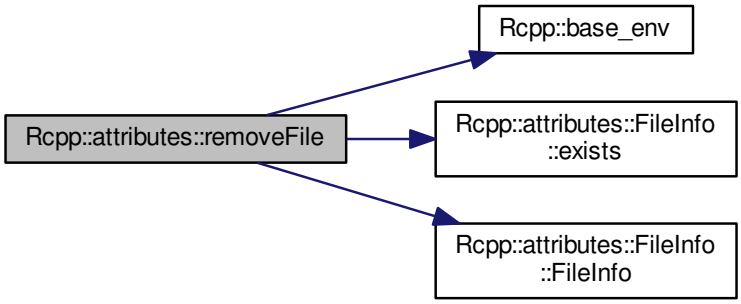
5.5.1.14 bool Rcpp::attributes::removeFile (const std::string & *path*)

Definition at line 2695 of file attributes.cpp.

References Rcpp::base_env(), Rcpp::attributes::FileInfo::exists(), and Rcpp::attributes::FileInfo::FileInfo().

Referenced by Rcpp::attributes::ExportsGenerator::remove().

Here is the call graph for this function:



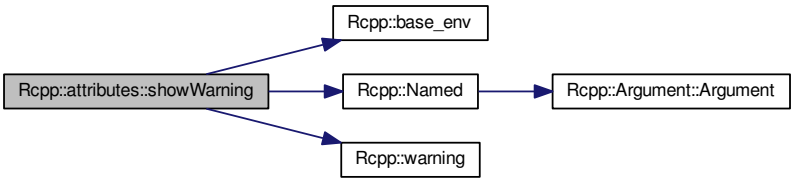
5.5.1.15 void Rcpp::attributes::showWarning (const std::string & *msg*)

Definition at line 2802 of file attributes.cpp.

References Rcpp::base_env(), Rcpp::Named(), and Rcpp::warning().

Referenced by Rcpp::attributes::SourceFileAttributesParser::attributeWarning(), compileAttributes(), and generateR↩ ArgList().

Here is the call graph for this function:



5.5.1.16 void Rcpp::attributes::stripQuotes (std::string * *pStr*)

Definition at line 2785 of file attributes.cpp.

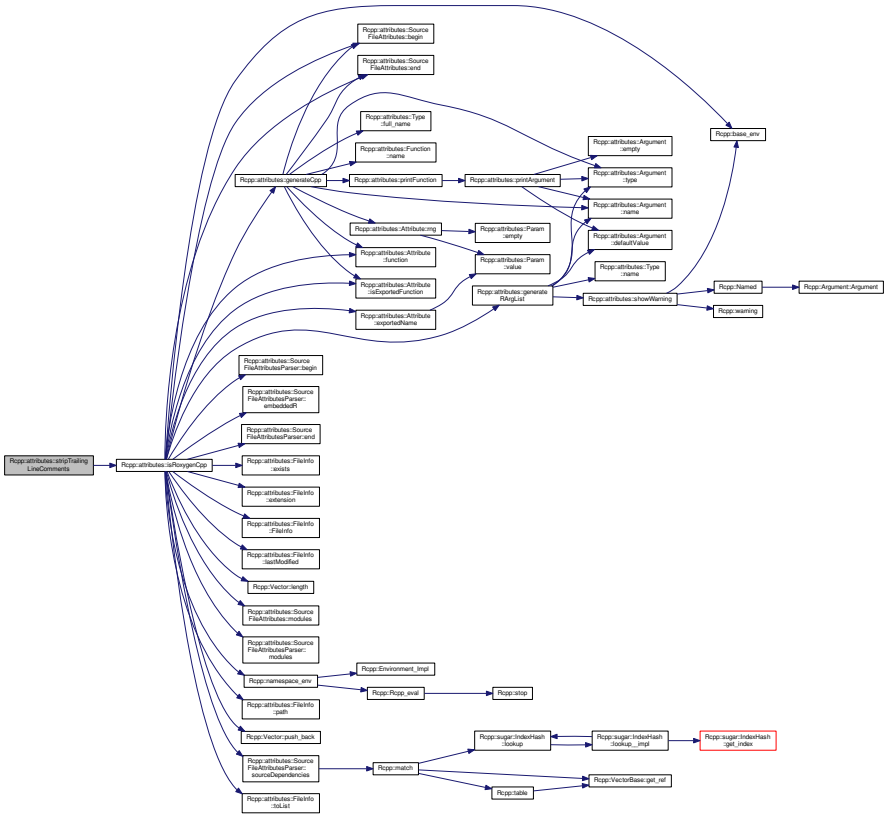
Referenced by Rcpp::attributes::Param::Param().

5.5.1.17 void Rcpp::attributes::stripTrailingLineComments (std::string * *pStr*)

Definition at line 2724 of file attributes.cpp.

References isRoxygenCpp().

Here is the call graph for this function:



5.5.1.18 void Rcpp::attributes::trimWhitespace (std::string * *pStr*)

Definition at line 2768 of file attributes.cpp.

Referenced by Rcpp::attributes::Param::Param(), Rcpp::attributes::SourceFileAttributesParser::parseAttribute(), Rcpp::attributes::SourceFileAttributesParser::parseFunction(), and Rcpp::attributes::SourceFileAttributesParser::parseType().

5.5.2 Variable Documentation

5.5.2.1 const char* const Rcpp::attributes::kDependsAttribute = "depends"

Definition at line 153 of file attributes.cpp.

Referenced by compileAttributes(), Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute(), and isRoxygenCpp().

5.5.2.2 const char* const Rcpp::attributes::kExportAttribute = "export"

Definition at line 150 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute(), isRoxygenCpp(), and Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.3 const char* const Rcpp::attributes::kExportName = "name"

Definition at line 151 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.4 const char* const Rcpp::attributes::kExportRng = "rng"

Definition at line 152 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.5 const char* const Rcpp::attributes::kInterfaceCpp = "cpp"

Definition at line 157 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::SourceFileAttributesParser::parseAttribute(), and Rcpp::attributes::ExportsGenerator::writeFunctions().

5.5.2.6 const char* const Rcpp::attributes::kInterfaceR = "r"

Definition at line 156 of file attributes.cpp.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::SourceFileAttributesParser::hasInterface(), and Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.7 const char* const Rcpp::attributes::kInterfacesAttribute = "interfaces"

Definition at line 155 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::hasInterface(), Rcpp::attributes::SourceFileAttributes↩Parser::isKnownAttribute(), and Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.8 const char* const Rcpp::attributes::kParamValueFalse = "false"

Definition at line 158 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.9 const char* const Rcpp::attributes::kParamValueFALSE = "FALSE"

Definition at line 160 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.10 const char* const Rcpp::attributes::kParamValueTrue = "true"

Definition at line 159 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute().

5.5.2.11 const char* const Rcpp::attributes::kParamValueTRUE = "TRUE"

Definition at line 161 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseAttribute(), and Rcpp::attributes::Attribute::rng().

5.5.2.12 const char* const Rcpp::attributes::kPluginsAttribute = "plugins"

Definition at line 154 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute(), and isRoxygenCpp().

5.5.2.13 const char *const Rcpp::attributes::kWhitespaceChars = " \f\n\r\t\v"

Definition at line 2715 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseFunction().

5.6 Rcpp::internal Namespace Reference

internal implementation details

Classes

- class const_generic_proxy
- class const_string_proxy
- class DimNameProxy
- class element_converter
- class generic_element_converter
- class generic_name_proxy
- class generic_proxy
- class InterruptedException
- class LazyVector
- class LazyVector< Rcpp::Vector< RTYPE > >
- class NamedPlaceHolder
- class RangeIndexer
- class simple_name_proxy
- class string_element_converter
- class string_name_proxy
- class string_proxy

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)
- 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)
- 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 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)

- 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)
- 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)
- std::ostream & operator<< (std::ostream &os, const const_string_proxy< STRSXP > &proxy)
- std::string operator+ (const std::string &x, const const_string_proxy< STRSXP > &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)

5.6.1 Detailed Description

internal implementation details

5.6.2 Function Documentation

5.6.2.1 template<typename T > T Rcpp::internal::as (SEXP *x,* ::Rcpp::traits::r_type_primitive_tag)

Definition at line 40 of file as.h.

5.6.2.2 template<typename T > T Rcpp::internal::as (SEXP *x,* ::Rcpp::traits::r_type_string_tag)

Definition at line 63 of file as.h.

5.6.2.3 template<typename T > T Rcpp::internal::as (SEXP *x,* ::Rcpp::traits::r_type_RcppString_tag)

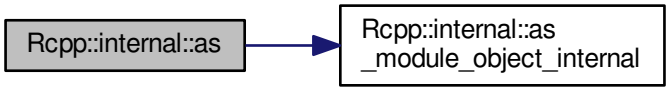
Definition at line 67 of file as.h.

5.6.2.4 template<typename T > T Rcpp::internal::as (SEXP *x,* ::Rcpp::traits::r_type_generic_tag)

Definition at line 77 of file as.h.

References as_module_object_internal(), DEMANGLE, and RCPP_DEBUG_1.

Here is the call graph for this function:



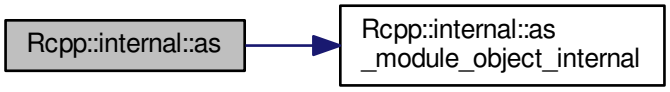
5.6.2.5 template<typename T > T Rcpp::internal::as (SEXP *x*, ::Rcpp::traits::r_type_module_object_const_pointer_tag)

handling object<T>

Definition at line 91 of file as.h.

References as_module_object_internal().

Here is the call graph for this function:



5.6.2.6 template<typename T > T Rcpp::internal::as (SEXP *x*, ::Rcpp::traits::r_type_module_object_pointer_tag)

Definition at line 96 of file as.h.

5.6.2.7 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 101 of file as.h.

5.6.2.8 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 107 of file as.h.

5.6.2.9 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 114 of file as.h.

5.6.2.10 template<typename T > T Rcpp::internal::as (SEXP *x*, ::Rcpp::traits::r_type_enum_tag)

handling enums by converting to int first

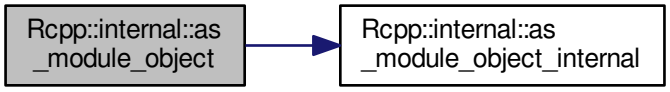
Definition at line 121 of file as.h.

5.6.2.11 template<typename T > object<T> Rcpp::internal::as_module_object (SEXP *x*)

Definition at line 86 of file as.h.

References as_module_object_internal().

Here is the call graph for this function:



5.6.2.12 void * Rcpp::internal::as_module_object_internal (SEXP *obj*) [inline]

Definition at line 24 of file as.h.

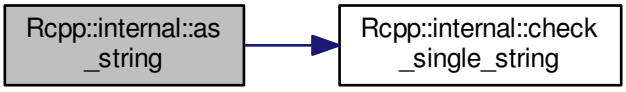
Referenced by as(), and as_module_object().

5.6.2.13 template<typename T > T Rcpp::internal::as_string (SEXP *x*, Rcpp::traits::true_type)

Definition at line 54 of file as.h.

References check_single_string().

Here is the call graph for this function:

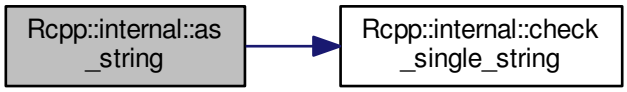


5.6.2.14 template<typename T > T Rcpp::internal::as_string (SEXP *x*, Rcpp::traits::false_type)

Definition at line 59 of file as.h.

References check_single_string().

Here is the call graph for this function:



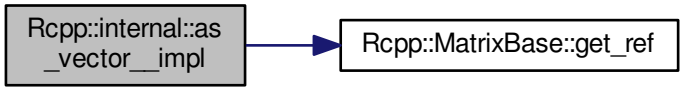
5.6.2.15 template<int RTYPE, bool NA, typename T > Rcpp::Vector<RTYPE> Rcpp::internal::as_vector__impl (MatrixBase< RTYPE, NA, T > & *t*, Rcpp::traits::false_type) [inline]

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:

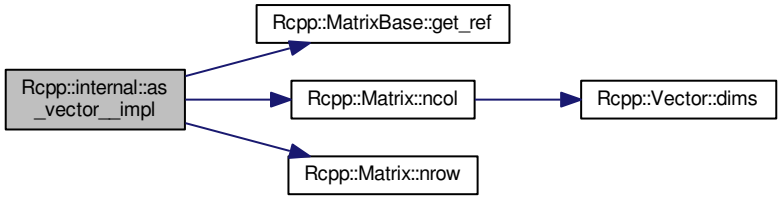


5.6.2.16 template<int RTYPE, bool NA, typename T > Rcpp::Vector<RTYPE> Rcpp::internal::as_vector__impl (MatrixBase< RTYPE, NA, T > & *t*, Rcpp::traits::true_type) [inline]

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 template<int RTYPE> SEXP Rcpp::internal::basic_cast (SEXP *x*)

Definition at line 50 of file r_cast.h.

Referenced by r_true_cast< CPLXSXP >(), r_true_cast< INTSXP >(), r_true_cast< LGLSXP >(), r_true_cast< R↩AWSXP >(), and r_true_cast< REALSXP >().

5.6.2.18 template<> Rcpp::Date Rcpp::internal::caster< double, Rcpp::Date > (double *from*) [inline]

Definition at line 126 of file Date.h.

5.6.2.19 template<> Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (double *from*) [inline]

Definition at line 99 of file Datetime.h.

5.6.2.20 template<> double Rcpp::internal::caster< Rcpp::Date, double > (Rcpp::Date *from*) [inline]

Definition at line 123 of file Date.h.

5.6.2.21 template<> double Rcpp::internal::caster< Rcpp::Datetime, double > (Rcpp::Datetime *from*) [inline]

Definition at line 96 of file Datetime.h.

5.6.2.22 const char* Rcpp::internal::check_single_string (SEXP *x*) [inline]

Definition at line 44 of file as.h.

Referenced by Rcpp::as< char >(), and as_string().

5.6.2.23 Rcomplex Rcpp::internal::complex__acos (Rcomplex *z*) [inline]

Definition at line 153 of file complex.h.

References complex__asin().

Referenced by complex__acosh().

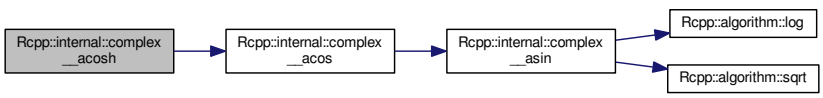
Here is the call graph for this function:

5.6.2.24 Rcomplex Rcpp::internal::complex__acosh (Rcomplex *z*) [inline]

Definition at line 186 of file complex.h.

References complex__acos().

Here is the call graph for this function:



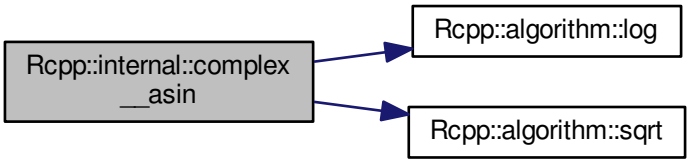
5.6.2.25 Rcomplex Rcpp::internal::complex__asin (Rcomplex *z*) [inline]

Definition at line 137 of file complex.h.

References Rcpp::algorithm::log(), RCPP_HYPOT, and Rcpp::algorithm::sqrt().

Referenced by complex__acos(), and complex__asinh().

Here is the call graph for this function:



5.6.2.26 Rcomplex Rcpp::internal::complex__asinh (Rcomplex *z*) [inline]

Definition at line 193 of file complex.h.

References complex__asin().

Here is the call graph for this function:

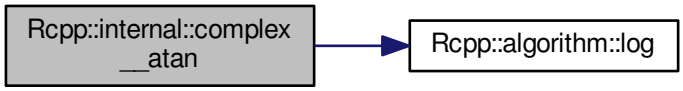
5.6.2.27 Rcomplex Rcpp::internal::complex__atan (Rcomplex *z*) [inline]

Definition at line 169 of file complex.h.

References Rcpp::algorithm::log().

Referenced by complex__atanh().

Here is the call graph for this function:

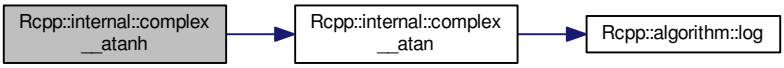


5.6.2.28 Rcomplex Rcpp::internal::complex__atanh (Rcomplex *z*) [inline]

Definition at line 203 of file complex.h.

References complex__atan().

Here is the call graph for this function:



5.6.2.29 Rcomplex Rcpp::internal::complex__Conj (Rcomplex *x*) [inline]

Definition at line 65 of file complex.h.

5.6.2.30 Rcomplex Rcpp::internal::complex__cos (Rcomplex *z*) [inline]

Definition at line 103 of file complex.h.

5.6.2.31 Rcomplex Rcpp::internal::complex__cosh (Rcomplex *z*) [inline]

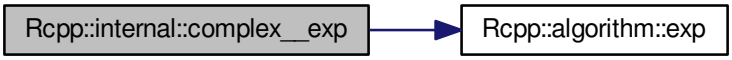
Definition at line 109 of file complex.h.

5.6.2.32 Rcomplex Rcpp::internal::complex__exp (Rcomplex *x*) [inline]

Definition at line 72 of file complex.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:



5.6.2.33 double Rcpp::internal::complex__Im (Rcomplex *x*) [inline]

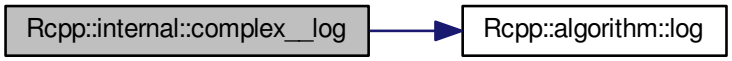
Definition at line 63 of file complex.h.

5.6.2.34 Rcomplex Rcpp::internal::complex__log (Rcomplex *x*) [inline]

Definition at line 79 of file complex.h.

References Rcpp::algorithm::log(), and RCPP_HYPOT.

Here is the call graph for this function:

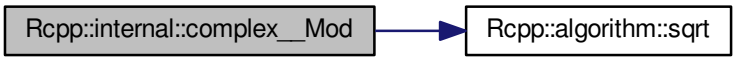


5.6.2.35 double Rcpp::internal::complex__Mod (Rcomplex *x*) [inline]

Definition at line 64 of file complex.h.

References Rcpp::algorithm::sqrt().

Here is the call graph for this function:



5.6.2.36 double Rcpp::internal::complex__Re (Rcomplex *x*) [inline]

Definition at line 62 of file complex.h.

5.6.2.37 Rcomplex Rcpp::internal::complex__sin (Rcomplex *z*) [inline]

Definition at line 115 of file complex.h.

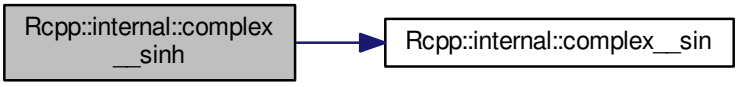
Referenced by complex__sinh().

5.6.2.38 Rcomplex Rcpp::internal::complex__sinh (Rcomplex *z*) [inline]

Definition at line 212 of file complex.h.

References complex__sin().

Here is the call graph for this function:

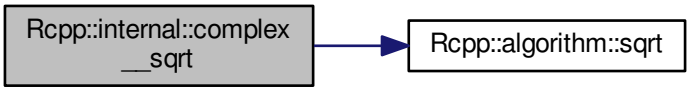


5.6.2.39 Rcomplex Rcpp::internal::complex__sqrt (Rcomplex *z*) [inline]

Definition at line 85 of file complex.h.

References RCPP_HYPOT, and Rcpp::algorithm::sqrt().

Here is the call graph for this function:



5.6.2.40 Rcomplex Rcpp::internal::complex__tan (Rcomplex *z*) [inline]

Definition at line 121 of file complex.h.

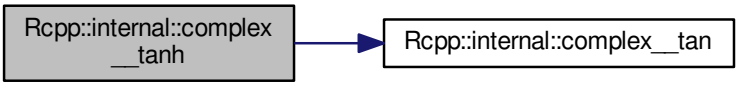
Referenced by complex__tanh().

5.6.2.41 Rcomplex Rcpp::internal::complex__tanh (Rcomplex *z*) [inline]

Definition at line 223 of file complex.h.

References complex__tan().

Here is the call graph for this function:



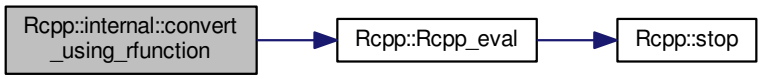
5.6.2.42 SEXP Rcpp::internal::convert_using_rfunction (SEXP *x*, const char *const *fun*) [inline]

Definition at line 30 of file r_cast.h.

References Rcpp::Rcpp_eval().

Referenced by Rcpp::Formula_Impl(), r_true_cast< EXPRSXP >(), r_true_cast< LANGSXP >(), r_true_cast< LIS↩ TSXP >(), r_true_cast< VECSXP >(), and Rcpp::DataFrame_Impl< StoragePolicy >::set__().

Here is the call graph for this function:

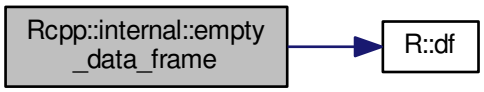


5.6.2.43 SEXP Rcpp::internal::empty_data_frame ( ) [inline]

Definition at line 28 of file DataFrame.h.

References R::df().

Here is the call graph for this function:



5.6.2.44 unsigned long Rcpp::internal::enterRNGScope ( ) [inline]

Definition at line 77 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::RNGScope::RNGScope().

5.6.2.45 unsigned long Rcpp::internal::exitRNGScope ( ) [inline]

Definition at line 83 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::RNGScope::~RNGScope().

5.6.2.46 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.

5.6.2.47 double Rcpp::internal::factorial (double *x*) [inline]

Definition at line 55 of file math.h.

Referenced by lfactorial().

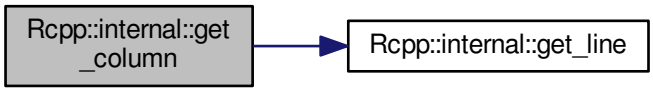
5.6.2.48 int Rcpp::internal::get_column (int *index,* int *nr*) [inline]

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:



5.6.2.49 int Rcpp::internal::get_column (int *index,* int *nr,* int *i*) [inline]

Definition at line 37 of file tools.h.

5.6.2.50 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.

5.6.2.51 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().

5.6.2.52 SEXP Rcpp::internal::get_Rcpp_namespace ( ) [inline]

Definition at line 95 of file routines.h.

References GET_CALLABLE.

Referenced by get_rcpp_cache(), Rcpp::Rcpp_namespace(), and registerFunctions().

5.6.2.53 char * Rcpp::internal::get_string_buffer ( ) [inline]

Definition at line 89 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.6.2.54 SEXP Rcpp::internal::getPosixClasses ( ) [inline]

Definition at line 154 of file Date.h.

Referenced by new_posixt_object(), and Rcpp::wrap_extra_steps< Rcpp::Datetime >().

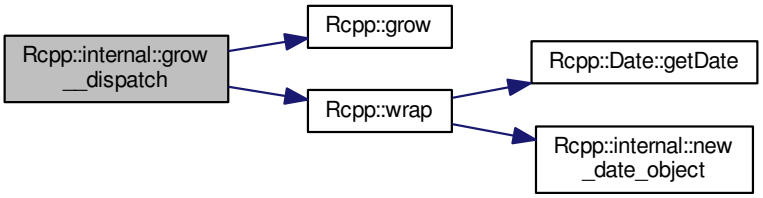
5.6.2.55 template<typename T > SEXP Rcpp::internal::grow__dispatch (::Rcpp::traits::false_type , const T & *head,* SEXP *tail*) [inline]

Definition at line 43 of file grow.h.

References Rcpp::grow(), and Rcpp::wrap().

Referenced by Rcpp::grow().

Here is the call graph for this function:

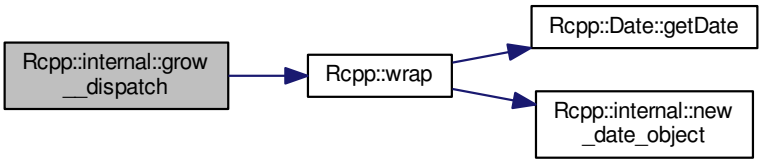


5.6.2.56 template<typename T > SEXP Rcpp::internal::grow__dispatch (::Rcpp::traits::true_type , const T & *head*, SEXP *tail*) [inline]

Definition at line 48 of file grow.h.

References Rcpp::wrap().

Here is the call graph for this function:



5.6.2.57 SEXP Rcpp::internal::interruptedError ( ) [inline]

Definition at line 40 of file Interrupt.h.

5.6.2.58 template<typename T > bool Rcpp::internal::is__dispatch (SEXP *x*, Rcpp::traits::false_type) [inline]

Definition at line 38 of file is.h.

5.6.2.59 template<typename T > bool Rcpp::internal::is__dispatch (SEXP *x*, Rcpp::traits::true_type) [inline]

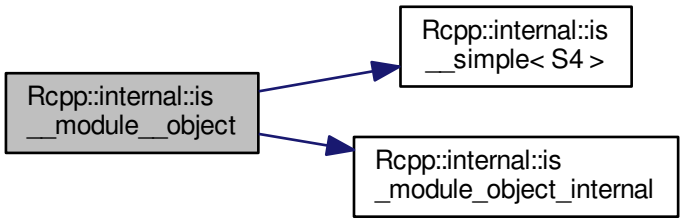
Definition at line 43 of file is.h.

5.6.2.60 template<typename T > bool Rcpp::internal::is__module__object (SEXP *x*)

Definition at line 158 of file is.h.

References is__simple< S4 >(), and is_module_object_internal().

Here is the call graph for this function:



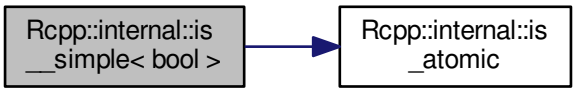
5.6.2.61 template<typename T > bool Rcpp::internal::is__simple (SEXP *x*)

5.6.2.62 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:

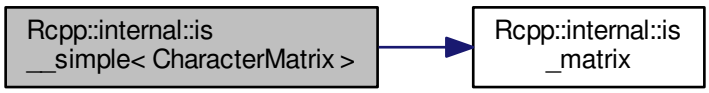


5.6.2.63 template<> bool **Rcpp::internal::is__simple**< **CharacterMatrix** > (SEXP *x*) [inline]

Definition at line 54 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.64 template<> bool Rcpp::internal::is__simple< CharacterVector > (SEXP *x*) [inline]

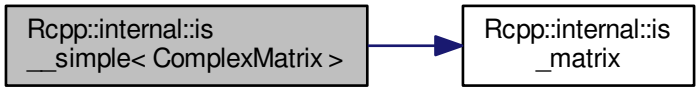
Definition at line 51 of file is.h.

5.6.2.65 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:



5.6.2.66 template<> bool Rcpp::internal::is__simple< ComplexVector > (SEXP *x*) [inline]

Definition at line 63 of file is.h.

5.6.2.67 template<> bool Rcpp::internal::is__simple< DataFrame > (SEXP *x*) [inline]

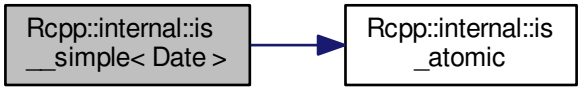
Definition at line 104 of file is.h.

5.6.2.68 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:

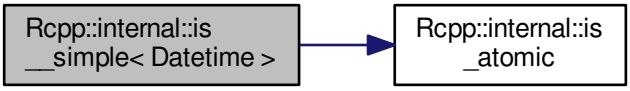


5.6.2.69 template<> bool Rcpp::internal::is__simple< Datetime > (SEXP *x*) [inline]

Definition at line 141 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.70 template<> bool Rcpp::internal::is__simple< DatetimeVector > (SEXP *x*) [inline]

Definition at line 147 of file is.h.

5.6.2.71 template<> bool Rcpp::internal::is__simple< DateVector > (SEXP *x*) [inline]

Definition at line 144 of file is.h.

5.6.2.72 template<> bool Rcpp::internal::is__simple< DottedPair > (SEXP *x*) [inline]

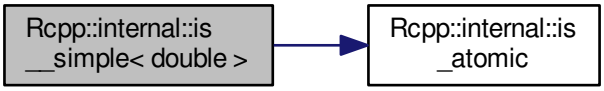
Definition at line 78 of file is.h.

5.6.2.73 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:



5.6.2.74 template<> bool Rcpp::internal::is__simple< Environment > (SEXP *x*) [inline]

Definition at line 130 of file is.h.

5.6.2.75 template<> bool Rcpp::internal::is__simple< Formula > (SEXP *x*) [inline]

Definition at line 133 of file is.h.

5.6.2.76 template<> bool Rcpp::internal::is__simple< Function > (SEXP *x*) [inline]

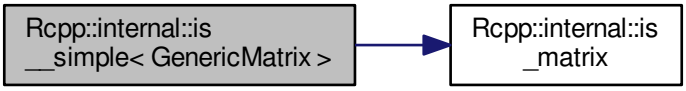
Definition at line 127 of file is.h.

5.6.2.77 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:



5.6.2.78 template<> bool Rcpp::internal::is__simple< int >(SEXP *x*) [inline]

Definition at line 33 of file is.h.

References is_atomic().

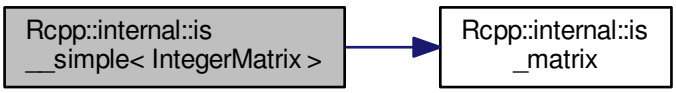
Here is the call graph for this function:

5.6.2.79 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:



5.6.2.80 template<> bool Rcpp::internal::is__simple< IntegerVector >(SEXP *x*) [inline]

Definition at line 60 of file is.h.

5.6.2.81 template<> bool Rcpp::internal::is__simple< Language >(SEXP *x*) [inline]

Definition at line 75 of file is.h.

5.6.2.82 template<> bool Rcpp::internal::is__simple< List >(SEXP *x*) [inline]

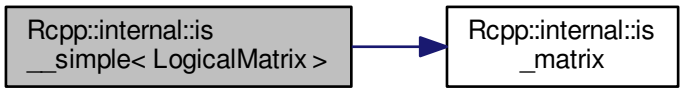
Definition at line 81 of file is.h.

5.6.2.83 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:



5.6.2.84 template<> bool Rcpp::internal::is__simple< LogicalVector >(SEXP *x*) [inline]

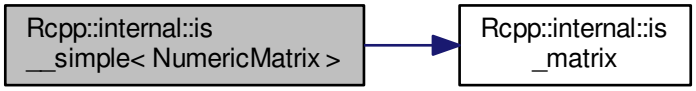
Definition at line 72 of file is.h.

5.6.2.85 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:



5.6.2.86 template<> bool Rcpp::internal::is__simple< NumericVector >(SEXP *x*) [inline]

Definition at line 69 of file is.h.

5.6.2.87 template<> bool Rcpp::internal::is__simple< Pairlist >(SEXP *x*) [inline]

Definition at line 124 of file is.h.

5.6.2.88 template<> bool Rcpp::internal::is__simple< Promise >(SEXP *x*) [inline]

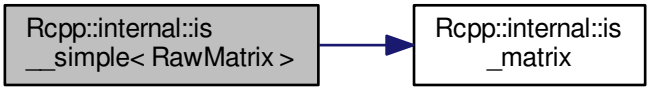
Definition at line 121 of file is.h.

5.6.2.89 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:



5.6.2.90 template<> bool Rcpp::internal::is__simple< RawVector >(SEXP x) [inline]

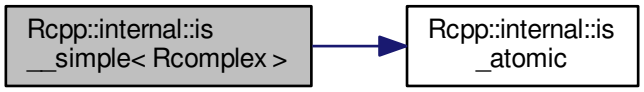
Definition at line 66 of file is.h.

5.6.2.91 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:



5.6.2.92 template<> bool Rcpp::internal::is__simple< Reference >(SEXP x) [inline]

Definition at line 117 of file is.h.

5.6.2.93 template<> bool Rcpp::internal::is__simple< RObject >(SEXP) [inline]

Definition at line 57 of file is.h.

5.6.2.94 template<> bool Rcpp::internal::is__simple< S4 >(SEXP *x*) [inline]

Definition at line 114 of file is.h.

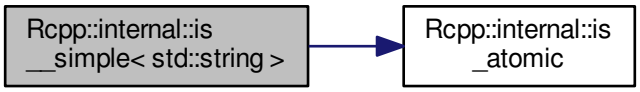
Referenced by is__module__object().

5.6.2.95 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:

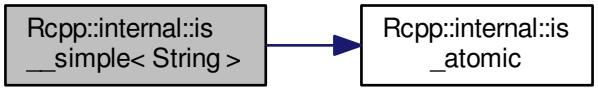


5.6.2.96 template<> bool Rcpp::internal::is__simple< String >(SEXP *x*) [inline]

Definition at line 45 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.97 template<> bool Rcpp::internal::is__simple< Symbol > (SEXP *x*) [inline]

Definition at line 111 of file is.h.

5.6.2.98 template<> bool Rcpp::internal::is__simple< WeakReference > (SEXP *x*) [inline]

Definition at line 108 of file is.h.

5.6.2.99 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 >().

5.6.2.100 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 >().

5.6.2.101 bool Rcpp::internal::is_module_object_internal (SEXP *obj,* const char * *clazz*) [inline]

Definition at line 151 of file is.h.

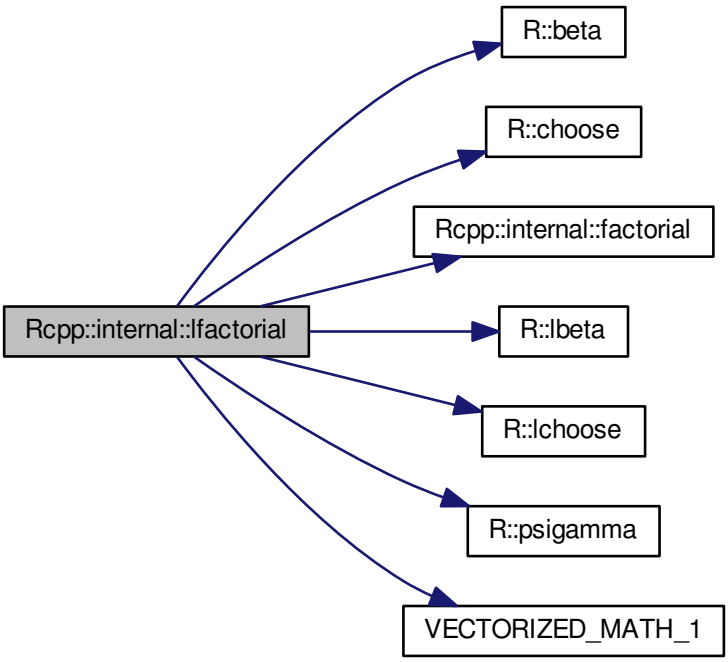
Referenced by is__module__object().

5.6.2.102 double Rcpp::internal::lfactorial (double *x*) [inline]

Definition at line 56 of file math.h.

References R::beta(), R::choose(), factorial(), R::lbeta(), R::lchoose(), R::psigamma(), SUGAR_BLOCK_2, and VEC↩ TORIZED_MATH_1().

Here is the call graph for this function:



5.6.2.103 template<> SEXP Rcpp::internal::make_charsexp< Rcpp::String > (const Rcpp::String & *s*) [inline]

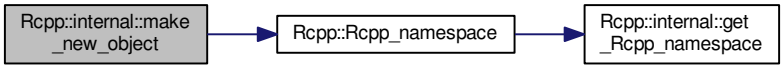
Definition at line 494 of file String.h.

5.6.2.104 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:



5.6.2.105 SEXP Rcpp::internal::new_date_object (double *d*) [inline]

Definition at line 167 of file Date.h.

Referenced by Rcpp::wrap().

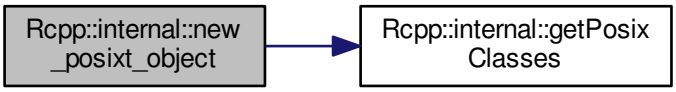
5.6.2.106 SEXP Rcpp::internal::new_posixt_object (double *d*) [inline]

Definition at line 161 of file Date.h.

References getPosixClasses().

Referenced by Rcpp::wrap< Datetime >().

Here is the call graph for this function:



5.6.2.107 std::string Rcpp::internal::operator+ (const std::string & *x*, const const_string_proxy< STRSXP > & *y*) [inline]

Definition at line 165 of file const_string_proxy.h.

5.6.2.108 std::string Rcpp::internal::operator+ (const std::string & *x*, const string_proxy< STRSXP > & *y*) [inline]

Definition at line 248 of file string_proxy.h.

5.6.2.109 template<int RT> bool Rcpp::internal::operator< (const const_string_proxy< RT > & *lhs*, const const_string_proxy< RT > & *rhs*)

Definition at line 127 of file const_string_proxy.h.

5.6.2.110 template<int RT> bool Rcpp::internal::operator< (const string_proxy< RT > & *lhs*, const string_proxy< RT > & *rhs*)

Definition at line 210 of file string_proxy.h.

5.6.2.111 std::ostream& Rcpp::internal::operator<< (std::ostream & *os,* const const_string_proxy< STRSXP > & *proxy*) [inline]

Definition at line 160 of file const_string_proxy.h.

5.6.2.112 std::ostream& Rcpp::internal::operator<< (std::ostream & *os,* const string_proxy< STRSXP > & *proxy*) [inline]

Definition at line 243 of file string_proxy.h.

5.6.2.113 template<int RT> bool Rcpp::internal::operator<= (const const_string_proxy< RT > & *lhs,* const const_string_proxy< RT > & *rhs*)

Definition at line 151 of file const_string_proxy.h.

5.6.2.114 template<int RT> bool Rcpp::internal::operator<= (const string_proxy< RT > & *lhs,* const string_proxy< RT > & *rhs*)

Definition at line 234 of file string_proxy.h.

5.6.2.115 template<int RT> bool Rcpp::internal::operator> (const const_string_proxy< RT > & *lhs,* const const_string_proxy< RT > & *rhs*)

Definition at line 135 of file const_string_proxy.h.

5.6.2.116 template<int RT> bool Rcpp::internal::operator> (const string_proxy< RT > & *lhs,* const string_proxy< RT > & *rhs*)

Definition at line 218 of file string_proxy.h.

5.6.2.117 template<int RT> bool Rcpp::internal::operator>= (const const_string_proxy< RT > & *lhs,* const const_string_proxy< RT > & *rhs*)

Definition at line 143 of file const_string_proxy.h.

5.6.2.118 template<int RT> bool Rcpp::internal::operator>= (const string_proxy< RT > & *lhs,* const string_proxy< RT > & *rhs*)

Definition at line 226 of file string_proxy.h.

5.6.2.119 template<typename T > T Rcpp::internal::primitive_as (SEXP *x*)

Definition at line 31 of file as.h.

5.6.2.120 template<int TARGET> SEXP Rcpp::internal::r_true_cast (SEXP *x*)

Definition at line 44 of file r_cast.h.

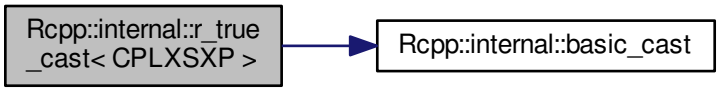
Referenced by Rcpp::r_cast().

5.6.2.121 template<> SEXP Rcpp::internal::r_true_cast< CPLXSXP > (SEXP *x*) [inline]

Definition at line 78 of file r_cast.h.

References basic_cast().

Here is the call graph for this function:

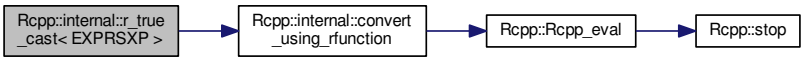


5.6.2.122 template<> SEXP Rcpp::internal::r_true_cast< EXPRSXP > (SEXP *x*) [inline]

Definition at line 115 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:

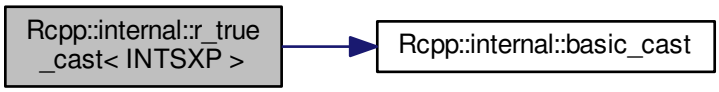


5.6.2.123 template<> SEXP Rcpp::internal::r_true_cast< INTSXP >(SEXP *x*) [inline]

Definition at line 66 of file r_cast.h.

References basic_cast().

Here is the call graph for this function:

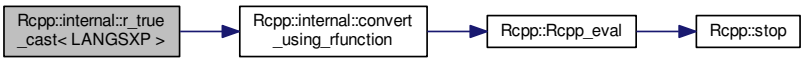


5.6.2.124 template<> SEXP Rcpp::internal::r_true_cast< LANGSXP >(SEXP *x*) [inline]

Definition at line 132 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:

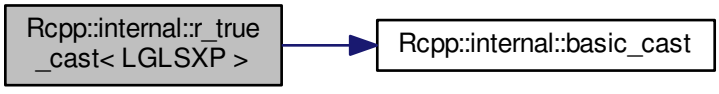


5.6.2.125 template<> SEXP Rcpp::internal::r_true_cast< LGLSXP >(SEXP *x*) [inline]

Definition at line 82 of file r_cast.h.

References basic_cast().

Here is the call graph for this function:

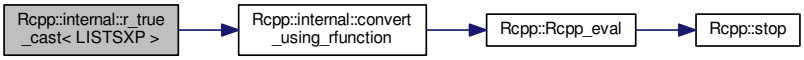


5.6.2.126 template<> SEXP Rcpp::internal::r_true_cast< LISTSXP > (SEXP *x*) [inline]

Definition at line 119 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:

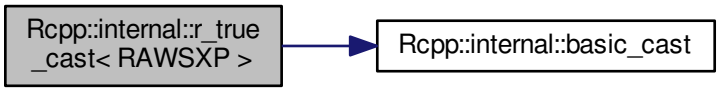


5.6.2.127 template<> SEXP Rcpp::internal::r_true_cast< RAWSXP > (SEXP *x*) [inline]

Definition at line 74 of file r_cast.h.

References basic_cast().

Here is the call graph for this function:

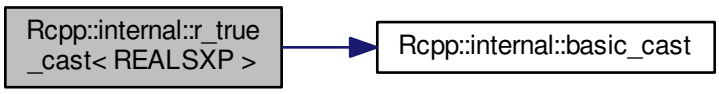


5.6.2.128 template<> SEXP Rcpp::internal::r_true_cast< REALSXP > (SEXP *x*) [inline]

Definition at line 70 of file r_cast.h.

References basic_cast().

Here is the call graph for this function:

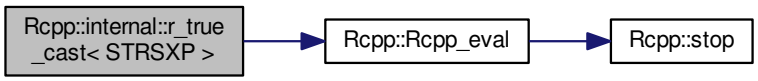


5.6.2.129 template<> SEXP Rcpp::internal::r_true_cast< STRSXP >(SEXP *x*) [inline]

Definition at line 87 of file r_cast.h.

References Rcpp::Rcpp_eval().

Here is the call graph for this function:

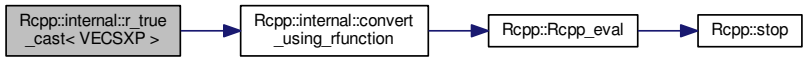


5.6.2.130 template<> SEXP Rcpp::internal::r_true_cast< VECSXP >(SEXP *x*) [inline]

Definition at line 111 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:

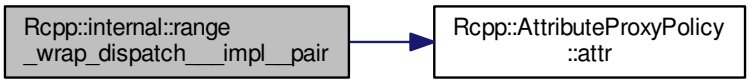


5.6.2.131 template<typename InputIterator , typename KEY , typename VALUE , int RTYPE> SEXP Rcpp::internal::range_wrap_dispatch___impl__pair (InputIterator *first,* InputIterator *last,* Rcpp::traits::true_type) [inline]

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:

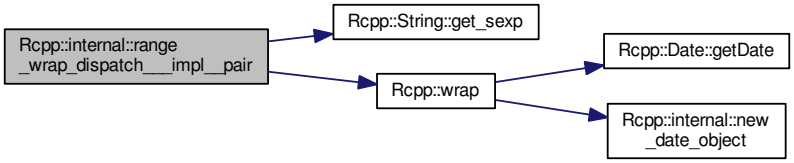


5.6.2.132 template<typename InputIterator , typename KEY , typename VALUE , int RTYPE> SEXP Rcpp::internal::range_wrap_dispatch__impl__pair (InputIterator *first,* InputIterator *last,* Rcpp::traits::false_type) [inline]

Definition at line 47 of file wrap.h.

References Rcpp::String::get_sexp(), and Rcpp::wrap().

Here is the call graph for this function:

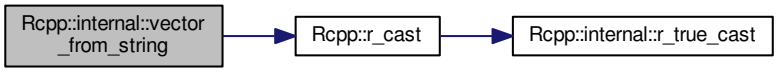


5.6.2.133 template<int RTYPE> SEXP Rcpp::internal::vector_from_string (const std::string & *st*) [inline]

Definition at line 29 of file vector_from_string.h.

References Rcpp::r_cast().

Here is the call graph for this function:



5.6.2.134 template<> SEXP Rcpp::internal::vector_from_string< EXPRSXP > (const std::string & *st*) [inline]

Definition at line 50 of file vector_from_string.h.

5.6.2.135 template<int RTYPE> SEXP Rcpp::internal::vector_from_string_expr (const std::string & *code*)

Definition at line 35 of file vector_from_string.h.

5.6.2.136 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.

5.7 Rcpp::InternalFunctionWithStdFunction Namespace Reference

Classes

- class CppFunctionBaseFromStdFunction
- class CppFunctionBaseFromStdFunction< void, Args... >

5.8 Rcpp::stats Namespace Reference

Classes

- class BetaGenerator
- class BinomGenerator
- class CauchyGenerator
- class CauchyGenerator_0
- class CauchyGenerator_1
- class ChisqGenerator
- class D0
- class D1
- class D2
- class D3
- class ExpGenerator
- class ExpGenerator__rate1
- class FGenerator_Finite_Finite
- class FGenerator_Finite_NotFinite
- class FGenerator_NotFinite_Finite
- class GammaGenerator
- class GeomGenerator
- class HyperGenerator
- class LNormGenerator
- class LNormGenerator_0
- class LNormGenerator_1
- class LogisGenerator
- class LogisGenerator_0
- class LogisGenerator_1
- class NBinomGenerator
- class NBinomGenerator_Mu
- class NChisqGenerator
- class NormGenerator

- class NormGenerator__mean0
- class NormGenerator__mean0__sd1
- class NormGenerator__sd1
- class P0
- class P1
- class P2
- class P3
- class PoissonGenerator
- class Q0
- class Q1
- class Q2
- class Q3
- 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 alph, 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)

5.8.1 Function Documentation

5.8.1.1 double Rcpp::stats::d_exp_0 (double *x,* int *give_log*) [inline]

Definition at line 30 of file exp.h.

References Rcpp::algorithm::exp(), and R_D__0.

Referenced by p_exp_0().

Here is the call graph for this function:

5.8.1.2 double Rcpp::stats::dcauchy_0 (double *x,* int *give_log*) [inline]

Definition at line 28 of file cauchy.h.

Referenced by qcauchy_1().

5.8.1.3 double Rcpp::stats::dcauchy_1 (double *x,* double *location,* int *give_log*) [inline]

Definition at line 32 of file cauchy.h.

Referenced by qcauchy_1().

5.8.1.4 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.

Referenced by qgamma_1().

Here is the call graph for this function:

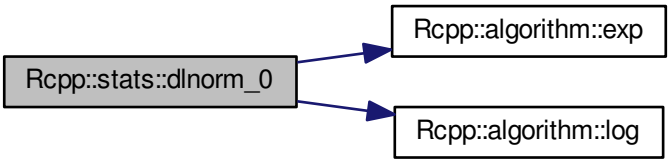
5.8.1.5 double Rcpp::stats::dlnorm_0 (double *x*, int *log_p*) [inline]

Definition at line 30 of file lnorm.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), and R_D__0.

Referenced by qlnorm_1().

Here is the call graph for this function:



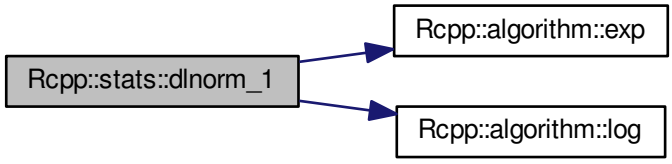
5.8.1.6 double Rcpp::stats::dlnorm_1 (double *x*, double *meanlog*, int *log_p*) [inline]

Definition at line 47 of file lnorm.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), and R_D__0.

Referenced by qlnorm_1().

Here is the call graph for this function:



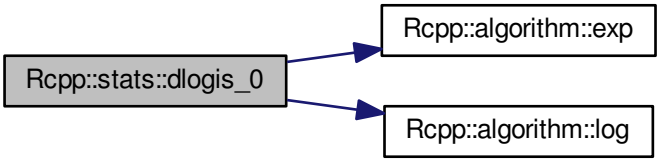
5.8.1.7 double Rcpp::stats::dlogis_0 (double *x*, int *give_log*) [inline]

Definition at line 28 of file logis.h.

References Rcpp::algorithm::exp(), and Rcpp::algorithm::log().

Referenced by qlogis_1().

Here is the call graph for this function:



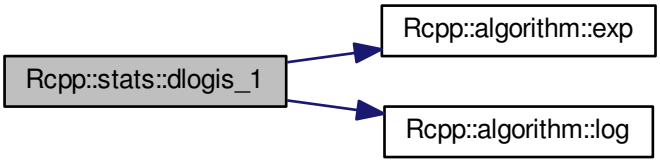
5.8.1.8 double Rcpp::stats::dlogis_1 (double *x*, double *location*, int *give_log*) [inline]

Definition at line 40 of file logis.h.

References Rcpp::algorithm::exp(), and Rcpp::algorithm::log().

Referenced by qlogis_1().

Here is the call graph for this function:



5.8.1.9 double Rcpp::stats::dnorm_0 (double *x*, int *give_log*) [inline]

Definition at line 43 of file norm.h.

References Rcpp::algorithm::exp(), and R_D__0.

Referenced by qnorm_0().

Here is the call graph for this function:

5.8.1.10 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(), ML_NAN, and R_D__0.

Referenced by qnorm_0().

Here is the call graph for this function:

5.8.1.11 double Rcpp::stats::dunif_0 (double *x*, int *give_log*) [inline]

Definition at line 34 of file unif.h.

References R_D__0.

Referenced by qunif_0().

5.8.1.12 double Rcpp::stats::dunif_1 (double *x*, double *a*, int *give_log*) [inline]

Definition at line 30 of file unif.h.

Referenced by qunif_0().

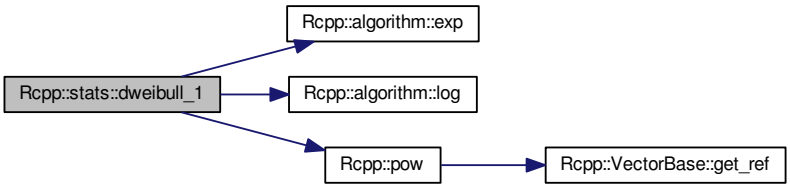
5.8.1.13 double Rcpp::stats::dweibull_1 (double *x*, double *shape*, int *give_log*) [inline]

Definition at line 30 of file weibull.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), ML_POSINF, Rcpp::pow(), and R_D__0.

Referenced by qweibull_1().

Here is the call graph for this function:

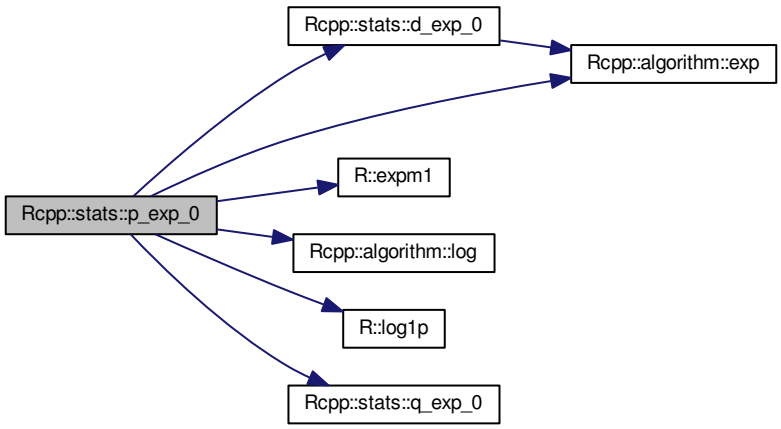


5.8.1.14 double Rcpp::stats::p_exp_0 (double *x*, int *lower_tail*, int *log_p*) [inline]

Definition at line 54 of file exp.h.

References d_exp_0(), Rcpp::algorithm::exp(), R::expm1(), Rcpp::algorithm::log(), R::log1p(), q_exp_0(), R_D_exp, R↩
_DT_0, and RCPP_DPQ_0.

Here is the call graph for this function:



5.8.1.15 double Rcpp::stats::pcauchy_0 (double *x*, int *lower_tail*, int *log_p*) [inline]

Definition at line 36 of file cauchy.h.

Referenced by qcauchy_1().

5.8.1.16 double Rcpp::stats::pcauchy_1 (double *x*, double *location*, int *lower_tail*, int *log_p*) [inline]

Definition at line 40 of file cauchy.h.

Referenced by qcauchy_1().

5.8.1.17 double Rcpp::stats::pgamma_1 (double *x*, double *alph*, int *lower_tail*, int *log_p*) [inline]

Definition at line 56 of file gamma.h.

Referenced by qgamma_1().

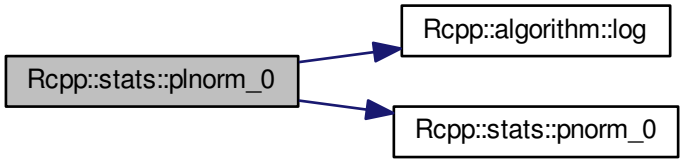
5.8.1.18 double Rcpp::stats::plnorm_0 (double *x*, int *lower_tail*, int *log_p*) [inline]

Definition at line 65 of file lnorm.h.

References Rcpp::algorithm::log(), pnorm_0(), and R_DT_0.

Referenced by qlnorm_1().

Here is the call graph for this function:



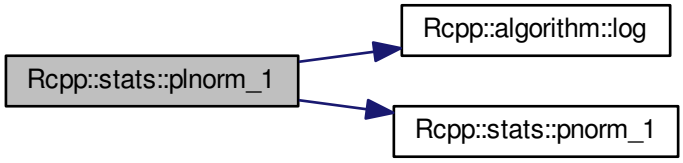
5.8.1.19 double Rcpp::stats::plnorm_1 (double *x*, double *meanlog*, int *lower_tail*, int *log_p*) [inline]

Definition at line 76 of file lnorm.h.

References Rcpp::algorithm::log(), pnorm_1(), and R_DT_0.

Referenced by qlnorm_1().

Here is the call graph for this function:



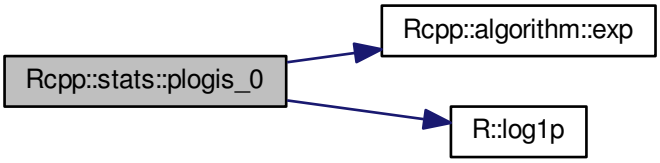
5.8.1.20 double Rcpp::stats::plogis_0 (double *x*, int *lower_tail*, int *log_p*) [inline]

Definition at line 54 of file logis.h.

References Rcpp::algorithm::exp(), R::log1p(), and R_P_bounds_Inf_01.

Referenced by qlogis_1().

Here is the call graph for this function:



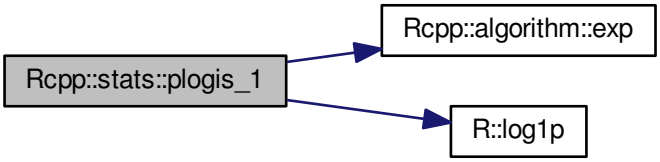
5.8.1.21 double Rcpp::stats::plogis_1 (double *x*, double *location*, int *lower_tail*, int *log_p*) [inline]

Definition at line 69 of file logis.h.

References Rcpp::algorithm::exp(), R::log1p(), and R_P_bounds_Inf_01.

Referenced by qlogis_1().

Here is the call graph for this function:



5.8.1.22 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 plnorm_0(), and qnorm_0().

5.8.1.23 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(), and qnorm_0().

5.8.1.24 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.

Referenced by qunif_0().

5.8.1.25 double Rcpp::stats::punif_1 (double *x*, double *a*, int *lower_tail*, int *log_p*) [inline]

Definition at line 46 of file unif.h.

Referenced by qunif_0().

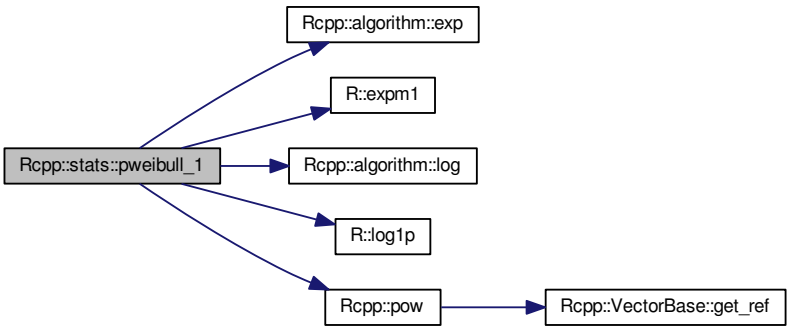
5.8.1.26 double Rcpp::stats::pweibull_1 (double *x*, double *shape*, int *lower_tail*, int *log_p*) [inline]

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.

Referenced by qweibull_1().

Here is the call graph for this function:



5.8.1.27 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.

Referenced by p_exp_0().

5.8.1.28 double Rcpp::stats::qcauchy_0 (double *p,* int *lower_tail,* int *log_p*) [inline]

Definition at line 44 of file cauchy.h.

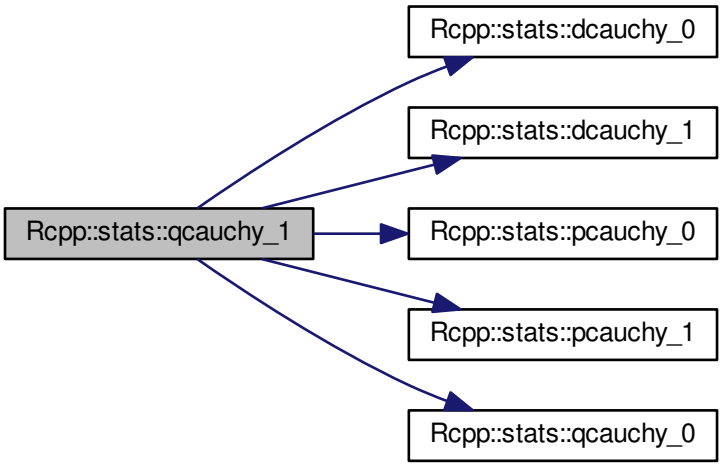
Referenced by qcauchy_1().

5.8.1.29 double Rcpp::stats::qcauchy_1 (double *p,* double *location,* int *lower_tail,* int *log_p*) [inline]

Definition at line 47 of file cauchy.h.

References dcauchy_0(), dcauchy_1(), pcauchy_0(), pcauchy_1(), qcauchy_0(), RCPP_DPQ_0, RCPP_DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:

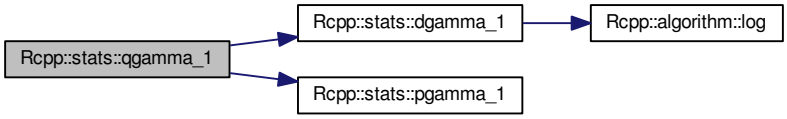


5.8.1.30 double Rcpp::stats::qgamma_1 (double *p,* double *alpha,* int *lower_tail,* int *log_p*) [inline]

Definition at line 59 of file gamma.h.

References dgamma_1(), pgamma_1(), RCPP_DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:



5.8.1.31 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.

Referenced by qlnorm_1().

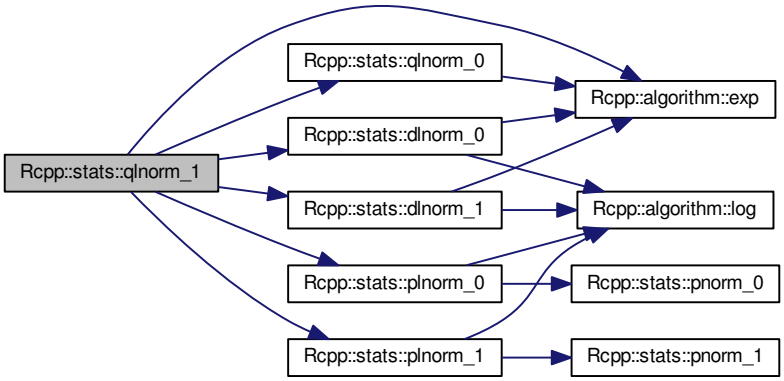
Here is the call graph for this function:

5.8.1.32 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 dlnorm_0(), dlnorm_1(), Rcpp::algorithm::exp(), ML_POSINF, plnorm_0(), plnorm_1(), qlnorm_0(), R_Q_↩
P01_boundaries, RCPP_DPQ_0, RCPP_DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:



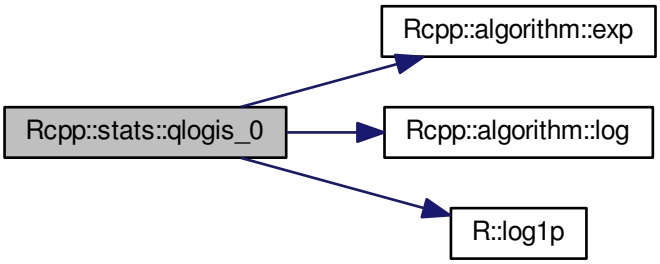
5.8.1.33 double Rcpp::stats::qlogis_0 (double *p*, int *lower_tail*, int *log_p*) [inline]

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.

Referenced by qlogis_1().

Here is the call graph for this function:

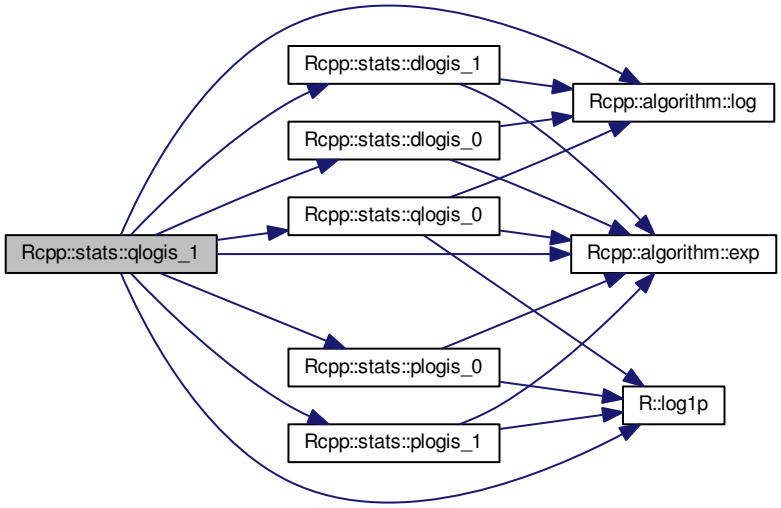


5.8.1.34 double Rcpp::stats::qlogis_1 (double *p,* double *location,* int *lower_tail,* int *log_p*) [inline]

Definition at line 106 of file logis.h.

References dlogis_0(), dlogis_1(), Rcpp::algorithm::exp(), Rcpp::algorithm::log(), R::log1p(), ML_NEGINF, ML_POSINF, plogis_0(), plogis_1(), qlogis_0(), R_Q_P01_boundaries, RCPP_DPQ_0, RCPP_DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:

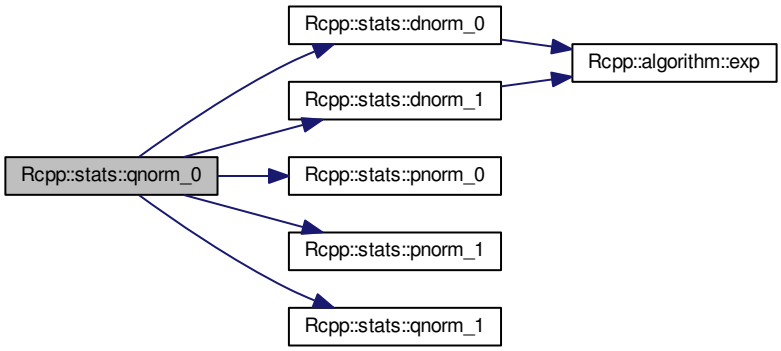


5.8.1.35 double Rcpp::stats::qnorm_0 (double *p,* int *lower_tail,* int *log_p*) [inline]

Definition at line 103 of file norm.h.

References dnorm_0(), dnorm_1(), pnorm_0(), pnorm_1(), qnorm_1(), RCPP_DPQ_0, RCPP_DPQ_1, and RCPP_↩ DPQ_2.

Here is the call graph for this function:



5.8.1.36 double Rcpp::stats::qnorm_1 (double *p*, double *mu*, int *lower_tail*, int *log_p*) [inline]

Definition at line 98 of file norm.h.

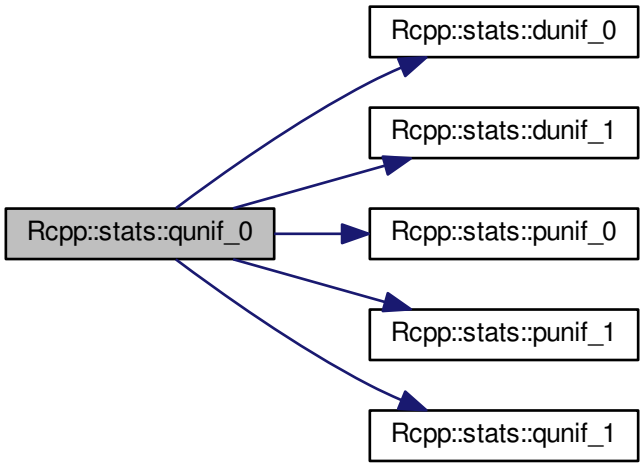
Referenced by qnorm_0().

5.8.1.37 double Rcpp::stats::qunif_0 (double *p*, int *lower_tail*, int *log_p*) [inline]

Definition at line 79 of file unif.h.

References dunif_0(), dunif_1(), punif_0(), punif_1(), qunif_1(), R_DT_qIv, R_Q_P01_check, RCPP_DPQ_0, RCPP_↩
DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:



5.8.1.38 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_qIv, and R_Q_P01_check.

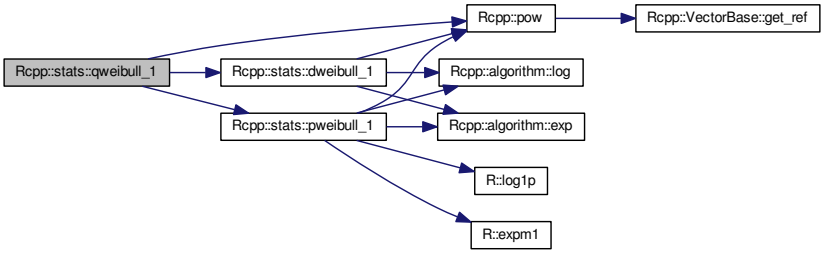
Referenced by qunif_0().

5.8.1.39 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 dweibull_1(), ML_POSINF, Rcpp::pow(), pweibull_1(), R_DT_Clog, R_Q_P01_boundaries, RCPP_DPQ_1, and RCPP_DPQ_2.

Here is the call graph for this function:



5.9 Rcpp::sugar Namespace Reference

Namespaces

- cbind_impl
- detail
- median_detail

Classes

- class All
- class All< false, T >
- class And_LogicalExpression_LogicalExpression
- class And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
- class And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class And_SingleLogicalResult_bool
- class And_SingleLogicalResult_SingleLogicalResult
- class And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class Any
- class Any< false, T >
- struct clamp_operator
- struct clamp_operator<REALSXP, true >
- class Clamp_Primitive_Vector_Primitive
- class Col
- class ColMeansImpl
- class ColMeansImpl< RTYPE, false, T, NA_RM >
- class ColMeansImpl< RTYPE, NA, T, true >
- class ColSumsImpl
- class ColSumsImpl< RTYPE, false, T, NA_RM >
- class ColSumsImpl< RTYPE, NA, T, true >
- class Comparator
- class Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >
- class Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >
- class Comparator_With_One_Value
- class Comparator_With_One_Value< RTYPE, Operator, false, T >
- class conversion_to_bool_is_forbidden
- class CountInserter
- class Cummax
- class Cummin
- class Cumprod
- class Cumsum
- class Diag_Extractor
- class Diag_Maker
- struct diag_result_type_trait
- class Diff
- class Diff< REALSXP, LHS_NA, LHS_T >

- class Diff< RTYPE, false, LHS_T >
- class Divides_Primitive_Vector
- class Divides_Primitive_Vector< REALSXP, false, T >
- class Divides_Primitive_Vector< REALSXP, NA, T >
- class Divides_Primitive_Vector< RTYPE, false, T >
- class Divides_Vector_Primitive
- class Divides_Vector_Primitive< REALSXP, false, T >
- class Divides_Vector_Primitive< REALSXP, NA, T >
- class Divides_Vector_Primitive< RTYPE, false, T >
- class Divides_Vector_Vector
- class Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class forbidden_conversion
- class forbidden_conversion< true >
- class Grabber
- class Head
- class IfElse
- class IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class IfElse_Primitive_Primitive
- class IfElse_Primitive_Primitive< RTYPE, false, COND_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 In
- class IndexHash
- class InSet
- class Intersect
- struct is_sugar_vector
- struct is_sugar_vector< Rcpp::Vector< RTYPE > >
- class IsFinite
- class IsInfinite
- class IsNa
- class IsNa< RTYPE, false, VEC_TYPE >
- class IsNa_Vector_is_na
- class IsNaN
- class Lapply
- class Lazy
- class LowerTri
- class Mapply_2
- class Mapply_2_Primitive_Vector
- class Mapply_2_Vector_Primitive
- class Mapply_3
- class Max
- class Max< RTYPE, false, T >
- class Mean

- class Mean< CPLXSXP, NA, T >
- class Mean< INTSXP, NA, T >
- class Mean< LGLSXP, NA, T >
- class Median
- class Median< RTYPE, false, T, NA_RM >
- class Median< RTYPE, NA, T, true >
- class Median< STRSXP, false, T, true >
- class Median< STRSXP, NA, T, NA_RM >
- class Median< STRSXP, NA, T, true >
- class Min
- class Min< RTYPE, false, T >
- class Minus_Primitive_Vector
- class Minus_Primitive_Vector< REALSXP, false, T >
- class Minus_Primitive_Vector< REALSXP, NA, T >
- class Minus_Primitive_Vector< RTYPE, false, T >
- class Minus_Vector_Primitive
- class Minus_Vector_Primitive< REALSXP, false, T >
- class Minus_Vector_Primitive< REALSXP, NA, T >
- class Minus_Vector_Primitive< RTYPE, false, T >
- class Minus_Vector_Vector
- class Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- struct negate
- struct negate< false >
- class Negate_SingleLogicalResult
- class Nona
- class Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >
- class NonaPrimitive
- class not_
- class not_< CPLXSXP, false >
- class not_< CPLXSXP, NA >
- class not_< REALSXP, false >
- class not_< REALSXP, NA >
- class not_< RTYPE, false >
- class Not_Vector
- class Or_LogicalExpression_LogicalExpression
- class Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
- class Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class Or_SingleLogicalResult_bool
- class Or_SingleLogicalResult_SingleLogicalResult
- class Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class Outer
- class Plus_Vector_Primitive

- class Plus_Vector_Primitive< REALSXP, false, T >
- class Plus_Vector_Primitive< REALSXP, NA, T >
- class Plus_Vector_Primitive< RTYPE, false, T >
- class Plus_Vector_Primitive_nona
- class Plus_Vector_Primitive_nona< REALSXP, false, T >
- class Plus_Vector_Primitive_nona< REALSXP, NA, T >
- class Plus_Vector_Primitive_nona< RTYPE, false, T >
- class Plus_Vector_Vector
- class Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- struct pmax_op
- struct pmax_op< INTSXP, LHS_NA, RHS_NA >
- struct pmax_op< REALSXP, false, false >
- struct pmax_op< REALSXP, false, true >
- struct pmax_op< REALSXP, true, false >
- struct pmax_op< REALSXP, true, true >
- class pmax_op_Vector_Primitive
- class pmax_op_Vector_Primitive< REALSXP, true >
- class Pmax_Vector_Primitive
- class Pmax_Vector_Vector
- struct pmin_op
- struct pmin_op< INTSXP, LHS_NA, RHS_NA >
- struct pmin_op< REALSXP, false, false >
- struct pmin_op< REALSXP, false, true >
- struct pmin_op< REALSXP, true, false >
- struct pmin_op< REALSXP, true, true >
- class pmin_op_Vector_Primitive
- class pmin_op_Vector_Primitive< REALSXP, true >
- class Pmin_Vector_Primitive
- class Pmin_Vector_Vector
- class Pow
- class Pow< INTSXP, false, T, EXPONENT_TYPE >
- class Pow< INTSXP, NA, T, EXPONENT_TYPE >
- class Range
- class Range< RTYPE, false, T >
- class RemoveFromSet
- class Rep
- class Rep_each
- class Rep_len
- class Rep_Single
- class Rev
- class Row
- class RowMeansImpl
- class RowMeansImpl< RTYPE, false, T, NA_RM >
- class RowMeansImpl< RTYPE, NA, T, true >
- class RowSumsImpl

- class RowSumsImpl< RTYPE, false, T, NA_RM >
- class RowSumsImpl< RTYPE, NA, T, true >
- class Sapply
- class Sapply< RTYPE, NA, T, Function, true >
- struct sapply_application_result_of
- class Sd
- class SelfHash
- class SelfInserter
- class SelfMatch
- class SeqLen
- class SetDiff
- class SetEqual
- class Sign
- class sign__impl
- class sign__impl< false, RTYPE >
- class SingleLogicalResult
- struct sugar_const_iterator_type
- struct sugar_const_iterator_type< CharacterVector >
- struct sugar_const_iterator_type< Rcpp::Vector< RTYPE > >
- class SugarBlock_1
- class SugarBlock_2
- class SugarBlock_2__PV
- class SugarBlock_2__VP
- class SugarBlock_3_VVV
- class SugarComplex
- class SugarIterator
- class SugarMath_1
- class SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >
- class SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >
- class Sum
- class Sum< REALSXP, NA, T >
- class Sum< RTYPE, false, T >
- class Table
- class Tail
- class Times_Vector_Primitive
- class Times_Vector_Primitive< REALSXP, false, T >
- class Times_Vector_Primitive< REALSXP, NA, T >
- class Times_Vector_Primitive< RTYPE, false, T >
- class Times_Vector_Primitive_nona
- class Times_Vector_Primitive_nona< REALSXP, false, T >
- class Times_Vector_Primitive_nona< REALSXP, NA, T >
- class Times_Vector_Primitive_nona< RTYPE, false, T >
- class Times_Vector_Vector
- class Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class unary_minus

- class unary_minus< CPLXSXP, false >
- class unary_minus< CPLXSXP, NA >
- class unary_minus< RTYPE, false >
- struct unary_minus_result_type
- struct unary_minus_result_type< LGLSXP >
- class UnaryMinus_Vector
- class Union
- class UpperTri
- class Var
- class Var< CPLXSXP, NA, T >
- class Vectorized
- class Vectorized_INTSXP
- class Vectorized_INTSXP< Func, false, VEC >
- class WhichMax
- class WhichMax< RTYPE, false, T >
- class WhichMin
- class WhichMin< RTYPE, false, T >

Typedefs

- typedef double(* DDFun) (double)

Functions

- template<typename T1 , typename T2 >
  cbind_impl::matrix_return< T1 >::type cbind (const T1 &t1, const T2 &t2)
- template<typename T1 , typename T2 , typename T3 >
  cbind_impl::matrix_return< T1 >::type 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 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 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 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 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 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)
- 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 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)
- 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 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)

- 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 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)
- 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 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)

- 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)

5.9.1 Typedef Documentation

5.9.1.1 typedef double(* Rcpp::sugar::DDFun) (double)

Definition at line 28 of file Vectorized_Math.h.

5.9.2 Function Documentation

5.9.2.1 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.

Referenced by cbind().

5.9.2.2 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.

5.9.2.3 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.

5.9.2.4 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.

5.9.2.5 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.

5.9.2.6 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.

5.9.2.7 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.

5.9.2.8 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.

5.9.2.9 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.

5.9.2.10 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.

5.9.2.11 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.

5.9.2.12 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.

5.9.2.13 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.

5.9.2.14 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.

5.9.2.15 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.

5.9.2.16 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.

5.9.2.17 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.

5.9.2.18 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.

5.9.2.19 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.

5.9.2.20 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.

5.9.2.21 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.

5.9.2.22 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.

5.9.2.23 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.

5.9.2.24 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.

5.9.2.25 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.

5.9.2.26 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.

5.9.2.27 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.

5.9.2.28 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.

5.9.2.29 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.

5.9.2.30 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.

5.9.2.31 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.

5.9.2.32 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.

5.9.2.33 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.

5.9.2.34 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.

5.9.2.35 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.

5.9.2.36 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.

5.9.2.37 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.

5.9.2.38 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.

5.9.2.39 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.

5.9.2.40 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.

5.9.2.41 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.

5.9.2.42 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.

5.9.2.43 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.

5.9.2.44 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.

5.9.2.45 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.

5.9.2.46 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.

5.9.2.47 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.

5.9.2.48 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.

5.9.2.49 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 cbind(), and MakeBindable.

Here is the call graph for this function:

5.9.2.50 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().

5.9.2.51 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().

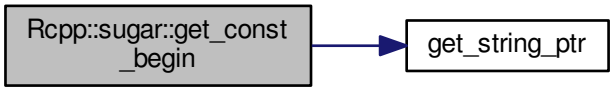
Here is the call graph for this function:

5.9.2.52 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:



5.9.2.53 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().

5.9.2.54 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.

5.9.2.55 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.

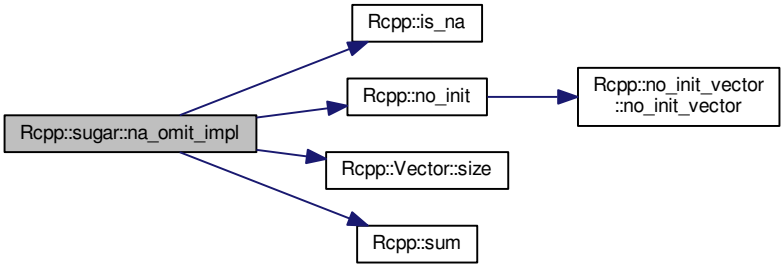
Referenced by Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect(), and Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Union().

5.9.2.56 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(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::sum().

Here is the call graph for this function:

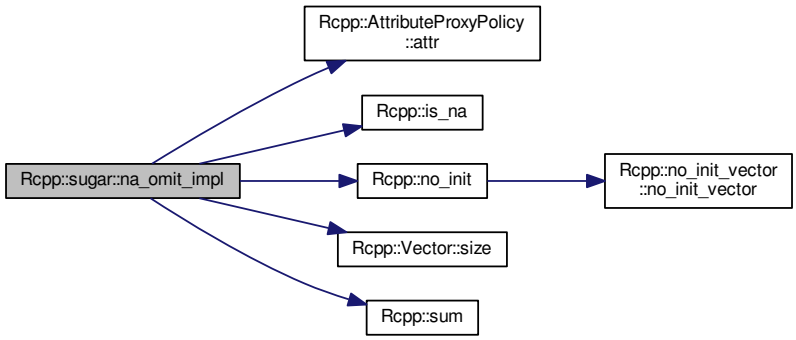


5.9.2.57 template<int RTYPE, bool NA, typename T > Vector<RTYPE> Rcpp::sugar::na_omit_impl (const T & *x*, Rcpp::traits::true_type)

Definition at line 44 of file na_omit.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::is_na(), Rcpp::no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::sum().

Here is the call graph for this function:



5.10 Rcpp::sugar::cbind_impl Namespace Reference

Namespaces

- detail

Classes

- class BindableExpression
- struct cbind_sexptype_traits
- struct cbind_sexptype_traits< SEXP >
- struct cbind_storage_type
- struct cbind_storage_type< LGLSXP >
- class ContainerBindable
- 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 >
- struct scalar
- class ScalarBindable

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)

5.10.1 Function Documentation

5.10.1.1 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()().

5.10.1.2 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.

5.10.1.3 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.

5.10.1.4 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.

5.10.1.5 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()().

5.10.1.6 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.

5.11 Rcpp::sugar::cbind_impl::detail Namespace Reference

Classes

- class has_stored_type
- struct MakeBindableCall
- struct MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >
- struct MakeBindableCall< Rcpp::Vector< LGLSXP >, true >
- struct MakeBindableCall< T, false >
- struct MakeBindableCall< T, true >
- struct matrix_return
- struct matrix_return< bool, false >
- struct matrix_return< Rcpp::Matrix< LGLSXP >, true >
- struct matrix_return< Rcpp::Vector< LGLSXP >, true >
- struct matrix_return< T, false >
- struct matrix_return< T, true >

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)

5.11.1 Function Documentation

5.11.1.1 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.

5.11.1.2 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.

5.12 Rcpp::sugar::detail Namespace Reference

Classes

- struct ColMeansReturn
- struct ColSumsReturn
- struct RowMeansReturn
- struct RowMeansReturn< CPLXSXP >
- struct RowSumsReturn
- struct RowSumsReturn< LGLSXP >

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)

5.12.1 Function Documentation

5.12.1.1 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().

5.12.1.2 bool Rcpp::sugar::detail::check_na (int *x*) [inline]

Definition at line 34 of file rowSums.h.

5.12.1.3 bool Rcpp::sugar::detail::check_na (Rboolean *x*) [inline]

Definition at line 38 of file rowSums.h.

5.12.1.4 bool Rcpp::sugar::detail::check_na (SEXP *x*) [inline]

Definition at line 42 of file rowSums.h.

5.12.1.5 bool Rcpp::sugar::detail::check_na (Rcomplex *x*) [inline]

Definition at line 46 of file rowSums.h.

5.12.1.6 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, false, T, false >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, false, T, false >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

5.12.1.7 void Rcpp::sugar::detail::div (Rcomplex * *lhs,* R_xlen_t *rhs*) [inline]

Definition at line 69 of file rowSums.h.

5.12.1.8 void Rcpp::sugar::detail::incr (double * *lhs,* double *rhs*) [inline]

Definition at line 51 of file rowSums.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::ColSumsImpl< RTY↩PE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, false, T, false >::get(), and Rcpp::sugar::Col↩MeansImpl< RTYPE, NA, T, true >::get().

5.12.1.9 void Rcpp::sugar::detail::incr (int * *lhs,* int *rhs*) [inline]

Definition at line 55 of file rowSums.h.

5.12.1.10 void Rcpp::sugar::detail::incr (Rcomplex * *lhs,* const Rcomplex & *rhs*) [inline]

Definition at line 59 of file rowSums.h.

5.12.1.11 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().

5.12.1.12 void Rcpp::sugar::detail::set_nan (Rcomplex * x) [inline]

Definition at line 79 of file rowSums.h.

5.13 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)

5.13.1 Function Documentation

5.13.1.1 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().

5.13.1.2 double Rcpp::sugar::median_detail::half (int *lhs*) [inline]

Definition at line 73 of file median.h.

5.13.1.3 Rcomplex Rcpp::sugar::median_detail::half (Rcomplex *lhs*) [inline]

Definition at line 77 of file median.h.

5.13.1.4 template<typename T > bool Rcpp::sugar::median_detail::less (T *lhs*, T *rhs*) [inline]

Definition at line 55 of file median.h.

5.13.1.5 template<> bool Rcpp::sugar::median_detail::less< Rcomplex > (Rcomplex *lhs*, Rcomplex *rhs*) [inline]

Definition at line 60 of file median.h.

5.14 Rcpp::traits Namespace Reference

traits used to dispatch wrap

Classes

- struct __sfinae_types
- class _has_iterator_helper
- class _has_matrix_interface_helper
- class _has_rtype_helper
- class _is_eigen_helper
- class _is_exporter_helper
- class _is_generator_helper
- class _is_importer_helper
- class _is_sugar_expression_helper
- struct both
- struct char_type
- struct char_type< const char * >
- struct char_type< const wchar_t * >
- struct container_exporter< Container, double >
- struct container_exporter< Container, int >
- class ContainerExporter
- struct enable_if
- struct enable_if< true, T >
- struct expands_to_logical
- struct expands_to_logical__impl
- struct expands_to_logical__impl< LGLSXP >
- struct Extractor
- struct has_iterator
- struct has_na
- struct has_na< CPLXSXP >
- struct has_na< INTSXP >

- struct has_na< LGLSXP >
- struct has_na< REALSXP >
- struct has_na< STRSXP >
- struct identity
- struct if_
- struct if_< false, LHS, RHS >
- struct init_type
- struct init_type< LGLSXP >
- struct init_type< STRSXP >
- struct input_parameter
- struct input_parameter< const T & >
- struct input_parameter< const T >
- struct input_parameter< T & >
- struct int2type
- struct integral_constant
- struct is_arithmetic
- struct is_arithmetic< const double >
- struct is_arithmetic< const float >
- struct is_arithmetic< const int >
- struct is_arithmetic< const long >
- struct is_arithmetic< const long double >
- struct is_arithmetic< const short >
- struct is_arithmetic< const unsigned int >
- struct is_arithmetic< const unsigned long >
- struct is_arithmetic< const unsigned short >
- struct is_arithmetic< double >
- struct is_arithmetic< float >
- struct is_arithmetic< int >
- struct is_arithmetic< long >
- struct is_arithmetic< long double >
- struct is_arithmetic< short >
- struct is_arithmetic< unsigned int >
- struct is_arithmetic< unsigned long >
- struct is_arithmetic< unsigned short >
- 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
- struct is_eigen_base
- struct is_exporter
- struct is_generator
- struct is_importer
- struct is_module_object
- struct is_named
- struct is_named< named_object< T > >
- struct is_named< Rcpp::Argument >

- struct is_pointer
- struct is_pointer< T * >
- struct is_primitive
- struct is_reference
- struct is_reference< _Tp & >
- struct is_sugar_expression
- struct is_trivial
- struct is_trivial< EXPRSXP >
- struct is_trivial< VECSXP >
- struct is_wide_string
- struct is_wide_string< char >
- struct is_wide_string< const char * >
- struct is_wide_string< const wchar_t * >
- struct is_wide_string< wchar_t >
- struct matrix_interface
- struct module_wrap_traits
- struct module_wrap_traits< T * >
- struct module_wrap_traits< void >
- class named_object
- class named_object< SEXP >
- struct needs_protection
- struct needs_protection< SEXP >
- struct normal_wrap_tag
- struct num2type
- struct pointer_wrap_tag
- struct proxy_based_const_iterator
- struct proxy_based_iterator
- class proxy_cache
- struct r_sexptype_needscast
- struct r_sexptype_needscast< double >
- struct r_sexptype_needscast< int >
- struct r_sexptype_needscast< Rbyte >
- struct r_sexptype_needscast< Rcomplex >
- struct r_sexptype_traits
- struct r_sexptype_traits< bool >
- struct r_sexptype_traits< const double >
- struct r_sexptype_traits< const int >
- struct r_sexptype_traits< double >
- struct r_sexptype_traits< float >
- struct r_sexptype_traits< int >
- struct r_sexptype_traits< long >
- struct r_sexptype_traits< long double >
- struct r_sexptype_traits< Rbyte >
- struct r_sexptype_traits< Rcomplex >
- struct r_sexptype_traits< Rcpp::Date >
- struct r_sexptype_traits< Rcpp::Datetime >
- struct r_sexptype_traits< Rcpp::String >
- struct r_sexptype_traits< short >
- struct r_sexptype_traits< std::complex< double > >
- struct r_sexptype_traits< std::complex< float > >
- struct r_sexptype_traits< std::string >

- struct r_sexptype_traits< unsigned int >
- struct r_sexptype_traits< unsigned long >
- struct r_sexptype_traits< unsigned short >
- struct r_type_enum_tag
- struct r_type_generic_tag
- struct r_type_module_object_const_pointer_tag
- struct r_type_module_object_const_reference_tag
- struct r_type_module_object_pointer_tag
- struct r_type_module_object_reference_tag
- struct r_type_module_object_tag
- struct r_type_pair_tag
- struct r_type_pairstring_generic_tag
- struct r_type_pairstring_primitive_tag
- struct r_type_pairstring_string_tag
- struct r_type_primitive_tag
- struct r_type_RcppString_tag
- struct r_type_string_tag
- struct r_type_traits
- struct r_type_traits< bool >
- struct r_type_traits< char >
- struct r_type_traits< const char * >
- struct r_type_traits< const double >
- struct r_type_traits< const int >
- struct r_type_traits< const wchar_t * >
- struct r_type_traits< double >
- struct r_type_traits< float >
- struct r_type_traits< int >
- struct r_type_traits< long >
- struct r_type_traits< long double >
- struct r_type_traits< Rbyte >
- struct r_type_traits< Rcomplex >
- struct r_type_traits< Rcpp::Date >
- struct r_type_traits< Rcpp::Datetime >
- struct r_type_traits< Rcpp::object< T > >
- struct r_type_traits< Rcpp::String >
- struct r_type_traits< short >
- struct r_type_traits< std::complex< double > >
- struct r_type_traits< std::complex< float > >
- struct r_type_traits< std::pair< const KEY, VALUE > >
- struct r_type_traits< std::pair< const std::string, bool > >
- struct r_type_traits< std::pair< const std::string, char > >
- 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, float > >
- struct r_type_traits< std::pair< const std::string, int > >
- struct r_type_traits< std::pair< const std::string, long > >
- struct r_type_traits< std::pair< const std::string, long 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, Rcpp::Date > >
- struct r_type_traits< std::pair< const std::string, Rcpp::Datetime > >

- struct r_type_traits< std::pair< const std::string, short > >
- struct r_type_traits< std::pair< const std::string, std::complex< double > > >
- struct r_type_traits< std::pair< const std::string, std::complex< float > > >
- 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, T > >
- struct r_type_traits< std::pair< const std::string, unsigned int > >
- struct r_type_traits< std::pair< const std::string, unsigned long > >
- struct r_type_traits< std::pair< const std::string, unsigned short > >
- struct r_type_traits< std::pair< const std::string, wchar_t > >
- struct r_type_traits< std::string >
- struct r_type_traits< std::wstring >
- struct r_type_traits< unsigned int >
- struct r_type_traits< unsigned long >
- struct r_type_traits< unsigned short >
- struct r_type_traits< wchar_t >
- class r_vector_cache
- struct r_vector_cache_type
- struct r_vector_cache_type< EXPRSXP, StoragePolicy >
- struct r_vector_cache_type< STRSXP, StoragePolicy >
- struct r_vector_cache_type< VECSXP, StoragePolicy >
- struct r_vector_const_iterator
- struct r_vector_const_iterator< EXPRSXP >
- struct r_vector_const_iterator< STRSXP >
- struct r_vector_const_iterator< VECSXP >
- struct r_vector_const_proxy
- struct r_vector_const_proxy< EXPRSXP >
- struct r_vector_const_proxy< STRSXP >
- struct r_vector_const_proxy< VECSXP >
- struct r_vector_element_converter
- struct r_vector_element_converter< EXPRSXP >
- struct r_vector_element_converter< STRSXP >
- struct r_vector_element_converter< VECSXP >
- struct r_vector_iterator
- struct r_vector_iterator< EXPRSXP >
- struct r_vector_iterator< STRSXP >
- struct r_vector_iterator< VECSXP >
- struct r_vector_name_proxy
- struct r_vector_name_proxy< EXPRSXP >
- struct r_vector_name_proxy< STRSXP >
- struct r_vector_name_proxy< VECSXP >
- struct r_vector_proxy
- struct r_vector_proxy< EXPRSXP >
- struct r_vector_proxy< STRSXP >
- struct r_vector_proxy< VECSXP >
- 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< CPLXSXP >
- struct storage_type< INTSXP >
- struct storage_type< LGLSXP >
- struct storage_type< RAWSXP >
- struct storage_type< REALSXP >
- struct un_pointer
- struct un_pointer< object< T > >
- struct un_pointer< T * >
- struct void_wrap_tag
- struct wrap_type_char_array
- struct wrap_type_enum_tag
- struct wrap_type_module_object_pointer_tag
- struct wrap_type_module_object_tag
- struct wrap_type_primitive_tag
- struct wrap_type_traits
- struct wrap_type_traits< bool >
- struct wrap_type_traits< char >
- struct wrap_type_traits< char[N]>
- struct wrap_type_traits< const char[N]>
- struct wrap_type_traits< const int >
- struct wrap_type_traits< double >
- struct wrap_type_traits< float >
- struct wrap_type_traits< int >
- struct wrap_type_traits< long >
- struct wrap_type_traits< long double >
- struct wrap_type_traits< Rbyte >
- struct wrap_type_traits< Rcomplex >
- struct wrap_type_traits< Rcpp::Date >
- struct wrap_type_traits< Rcpp::Datetime >
- struct wrap_type_traits< Rcpp::object< T > >
- struct wrap_type_traits< Rcpp::String >
- struct wrap_type_traits< short >
- struct wrap_type_traits< std::complex< double > >
- struct wrap_type_traits< std::complex< float > >
- struct wrap_type_traits< std::string >
- struct wrap_type_traits< std::wstring >
- struct wrap_type_traits< unsigned int >
- struct wrap_type_traits< unsigned long >
- struct wrap_type_traits< unsigned short >
- struct wrap_type_traits< wchar_t >
- struct wrap_type_unknown_tag

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)

5.14.1 Detailed Description

traits used to dispatch wrap

5.14.2 Typedef Documentation

5.14.2.1 typedef integral_constant<bool, false> Rcpp::traits::false_type

Definition at line 35 of file integral_constant.h.

5.14.2.2 typedef integral_constant<bool, true> Rcpp::traits::true_type

Definition at line 34 of file integral_constant.h.

5.14.3 Function Documentation

5.14.3.1 template<int RTYPE> storage_type<RTYPE>::type Rcpp::traits::get_na ( )

5.14.3.2 template<> Rcomplex Rcpp::traits::get_na< CPLXSXP > ( ) [inline]

Definition at line 42 of file get_na.h.

5.14.3.3 template<> int Rcpp::traits::get_na< INTSXP > ( ) [inline]

Definition at line 33 of file get_na.h.

5.14.3.4 template<> int Rcpp::traits::get_na< LGLSXP > ( ) [inline]

Definition at line 36 of file get_na.h.

5.14.3.5 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[]().

5.14.3.6 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< ST↩ RSXP, NA, T, true >::operator result_type(), and Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type().

5.14.3.7 template<> SEXP Rcpp::traits::get_na< VECSXP >( ) [inline]

Definition at line 54 of file get_na.h.

5.14.3.8 template<int RTYPE> bool Rcpp::traits::is_finite (typename storage_type< RTYPE >::type)

5.14.3.9 template<> bool Rcpp::traits::is_finite< CPLXSXP >(Rcomplex *x*) [inline]

Definition at line 44 of file is_finite.h.

5.14.3.10 template<> bool Rcpp::traits::is_finite< INTSXP >(int *x*) [inline]

Definition at line 34 of file is_finite.h.

5.14.3.11 template<> bool Rcpp::traits::is_finite< LGLSXP >(int *x*) [inline]

Definition at line 54 of file is_finite.h.

5.14.3.12 template<> bool Rcpp::traits::is_finite< REALSXP >(double *x*) [inline]

Definition at line 39 of file is_finite.h.

5.14.3.13 template<> bool Rcpp::traits::is_finite< STRSXP >(SEXP) [inline]

Definition at line 49 of file is_finite.h.

5.14.3.14 template<int RTYPE> bool Rcpp::traits::is_infinite (typename storage_type< RTYPE >::type)

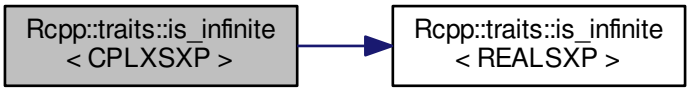
Definition at line 30 of file is_infinite.h.

5.14.3.15 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:



5.14.3.16 template<> bool Rcpp::traits::is_infinite< REALSXP > (double *x*) [inline]

Definition at line 35 of file is_infinite.h.

Referenced by is_infinite< CPLXSXP >().

5.14.3.17 template<int RTYPE> bool Rcpp::traits::is_na (typename storage_type< RTYPE >::type)

Definition at line 31 of file is_na.h.

5.14.3.18 template<> bool Rcpp::traits::is_na< CPLXSXP > (Rcomplex *x*) [inline]

Definition at line 46 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[]().

5.14.3.19 template<> bool Rcpp::traits::is_na< INTSXP > (int *x*) [inline]

Definition at line 36 of file is_na.h.

Referenced by Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[]().

5.14.3.20 template<> bool Rcpp::traits::is_na< LGLSXP >(int x) [inline]

Definition at line 57 of file is_na.h.

Referenced by Rcpp::sugar::negate< NA >::apply(), Rcpp::sugar::All< NA, T >::apply(), Rcpp::sugar::Any< NA, T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::SingleLogicalResult< false, Any< false, 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[]().

5.14.3.21 template<> bool Rcpp::traits::is_na< REALSXP >(double x) [inline]

Definition at line 41 of file is_na.h.

Referenced by Rcpp::sugar::not_< REALSXP, NA >::apply(), Rcpp::Datetime::is_na(), Rcpp::Date::is_na(), Rcpp::sugar::pmax_op< REALSXP, true, true >::operator()(), Rcpp::sugar::pmin_op< REALSXP, true, true >::operator()(), Rcpp::sugar::clamp_operator<REALSXP, true >::operator()(), Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::operator()(), and Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::operator()().

5.14.3.22 template<> bool Rcpp::traits::is_na< STRSXP >(SEXP x) [inline]

Definition at line 52 of file is_na.h.

5.14.3.23 template<int RTYPE> bool Rcpp::traits::is_nan (typename storage_type< RTYPE >::type)

Definition at line 30 of file is_nan.h.

5.14.3.24 template<> bool Rcpp::traits::is_nan< CPLXSXP >(Rcomplex x) [inline]

Definition at line 40 of file is_nan.h.

5.14.3.25 template<> bool Rcpp::traits::is_nan< REALSXP >(double x) [inline]

Definition at line 35 of file is_nan.h.

5.15 std Namespace Reference

5.16 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)

5.16.1 Typedef Documentation

5.16.1.1 typedef const FormatList& tinyformat::FormatListRef

Reference to type-opaque format list for passing to vformat()

Definition at line 858 of file tinyformat.h.

5.16.2 Function Documentation

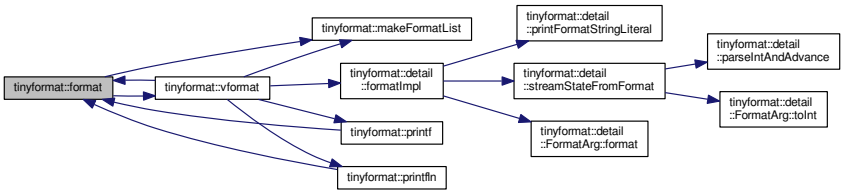
5.16.2.1 void tinyformat::format (std::ostream & *out,* const char * *fmt*) [inline]

Definition at line 991 of file tinyformat.h.

References makeFormatList(), and vformat().

Referenced by format(), printf(), printfln(), Rcpp::stop(), vformat(), and Rcpp::warning().

Here is the call graph for this function:

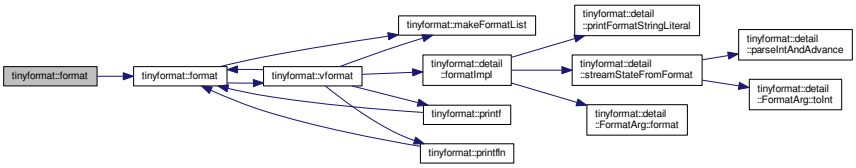


5.16.2.2 std::string tinyformat::format (const char * *fmt*) [inline]

Definition at line 996 of file tinyformat.h.

References format().

Here is the call graph for this function:



5.16.2.3 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.

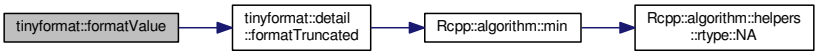
By default, formatValue() uses the usual stream insertion operator operator<< to format the type T, with special cases for the c and p conversions.

Definition at line 321 of file tinyformat.h.

References tinyformat::detail::formatTruncated().

Referenced by tinyformat::detail::FormatArg::formatImpl().

Here is the call graph for this function:



5.16.2.4 detail::FormatListN<0> tinyformat::makeFormatList ( ) [inline]

Definition at line 930 of file tinyformat.h.

Referenced by format(), tinyformat::detail::FormatListN< 0 >::FormatListN(), and vformat().

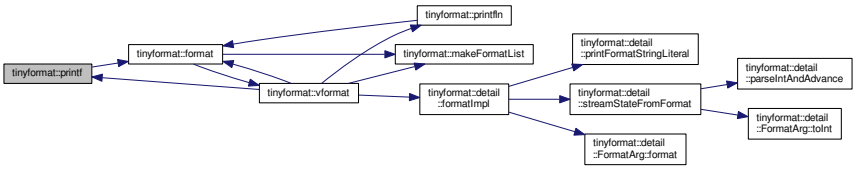
5.16.2.5 void tinyformat::printf (const char * *fmt*) [inline]

Definition at line 1003 of file tinyformat.h.

References format().

Referenced by vformat().

Here is the call graph for this function:



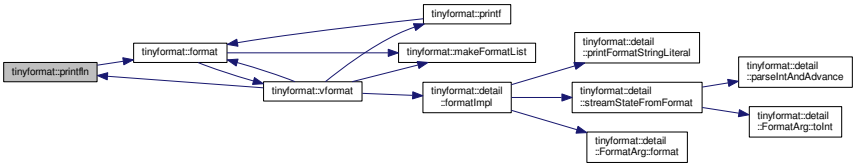
5.16.2.6 void tinyformat::printfln (const char * *fmt*) [inline]

Definition at line 1008 of file tinyformat.h.

References format().

Referenced by vformat().

Here is the call graph for this function:



5.16.2.7 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.

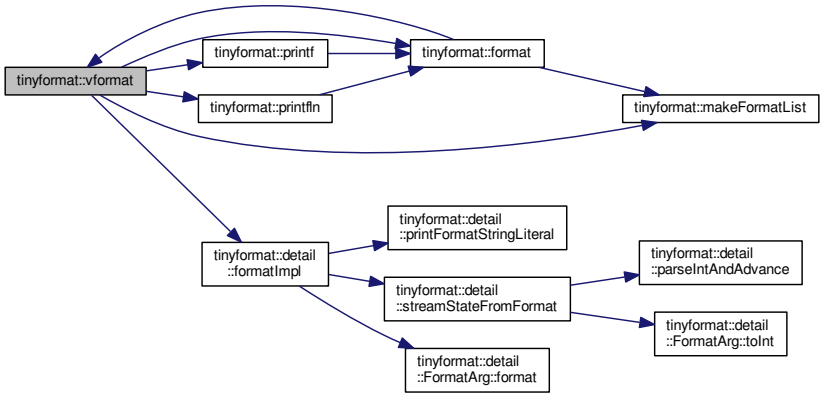
The name vformat() is chosen for the semantic similarity to vprintf(): the list of format arguments is held in a single function argument.

Definition at line 949 of file tinyformat.h.

References format(), tinyformat::detail::formatImpl(), tinyformat::FormatList::m_formatters, tinyformat::FormatList::m_N, makeFormatList(), printf(), and printfln().

Referenced by format(), and tinyformat::FormatList::FormatList().

Here is the call graph for this function:



5.17 tinyformat::detail Namespace Reference

Classes

- struct convertToInt
- struct convertToInt< T, true >
- class FormatArg
- class FormatListN
- class FormatListN< 0 >
- struct formatValueAsType
- struct formatValueAsType< T, fmtT, true >
- struct is_convertible
- struct is_wchar
- struct is_wchar< const wchar_t * >
- struct is_wchar< const wchar_t[n]>
- struct is_wchar< wchar_t * >
- struct is_wchar< wchar_t[n]>

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)

5.17.1 Function Documentation

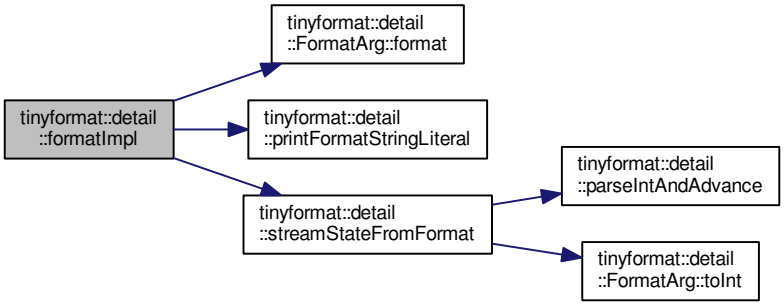
5.17.1.1 void tinyformat::detail::formatImpl (std::ostream & *out*, const char * *fmt*, const detail::FormatArg * *formatters*, int *numFormatters*) [inline]

Definition at line 776 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:



5.17.1.2 template<typename T > void tinyformat::detail::formatTruncated (std::ostream & *out*, const T & *value*, int *ntrunc*) [inline]

Definition at line 280 of file tinyformat.h.

References Rcpp::algorithm::min().

Referenced by tinyformat::formatValue().

Here is the call graph for this function:

5.17.1.3 int tinyformat::detail::parseIntAndAdvance (const char *& *c*) [inline]

Definition at line 547 of file tinyformat.h.

Referenced by streamStateFromFormat().

5.17.1.4 const char* tinyformat::detail::printFormatStringLiteral (std::ostream & *out,* const char * *fmt*) [inline]

Definition at line 561 of file tinyformat.h.

Referenced by formatImpl().

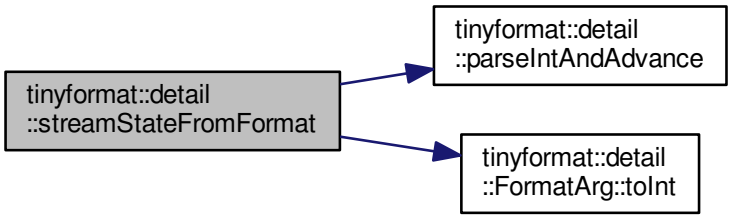
5.17.1.5 const char* tinyformat::detail::streamStateFromFormat (std::ostream & *out,* bool & *spacePadPositive,* int & *ntrunc,* const char * *fmtStart,* const detail::FormatArg * *formatters,* int & *argIndex,* int *numFormatters*) [inline]

Definition at line 593 of file tinyformat.h.

References parseIntAndAdvance(), TINYFORMAT_ERROR, and tinyformat::detail::FormatArg::toInt().

Referenced by formatImpl().

Here is the call graph for this function:



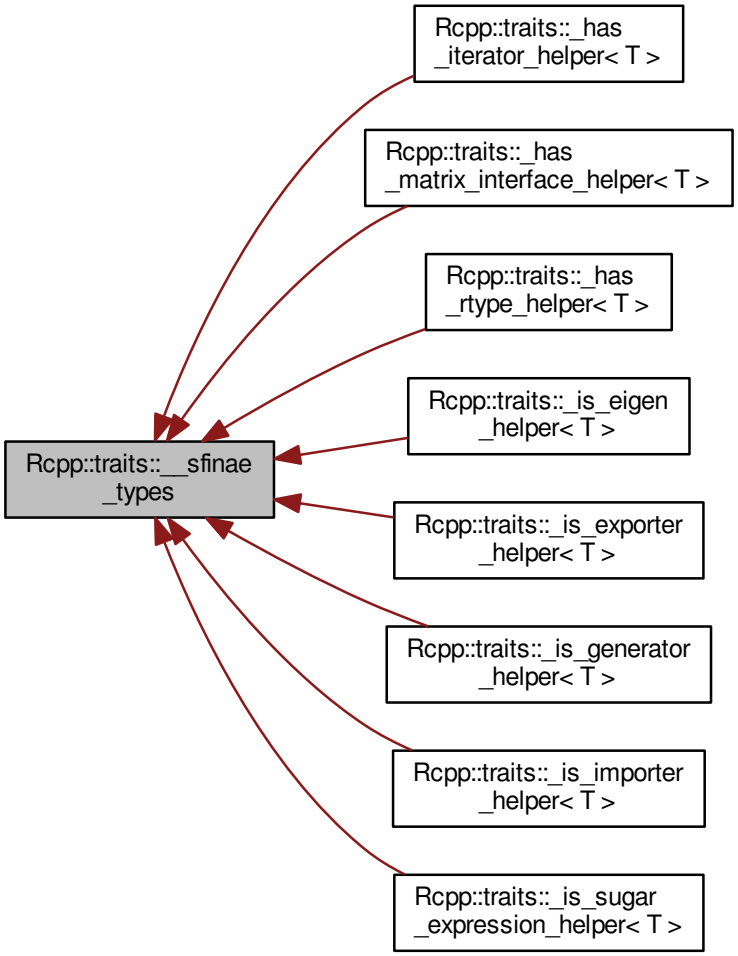
Chapter 6

Class Documentation

6.1 Rcpp::traits::__sfinae_types Struct Reference

`#include <has_iterator.h>`

Inheritance diagram for Rcpp::traits::__sfinae_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 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 char Rcpp::traits::__sfinae_types::__two::__arr[2]

Definition at line 36 of file has_iterator.h.

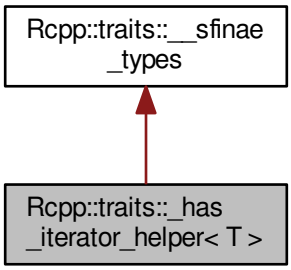
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

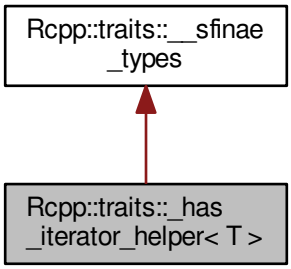
6.3 Rcpp::traits::_has_iterator_helper< T > Class Template Reference

`#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 template<typename T > template<typename U > static __one Rcpp::traits::_has_iterator_helper< T >::__test (_Wrap_type< typename U::iterator > *) [static], [private]

6.3.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_has_iterator_helper< T >::__test (...) [static], [private]

6.3.3 Member Data Documentation

6.3.3.1 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.

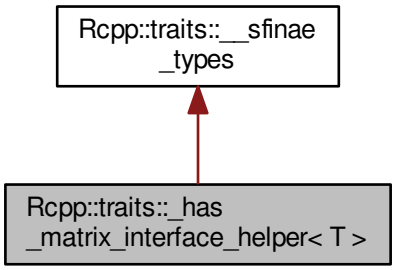
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

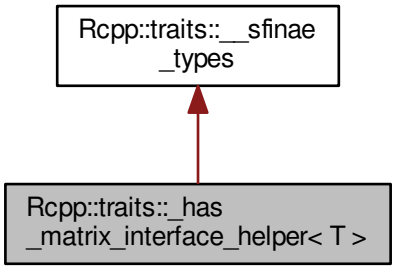
6.4 Rcpp::traits::_has_matrix_interface_helper< T > Class Template Reference

`#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 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]

6.4.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_has_matrix_interface_helper< T >::__test (...) [static], [private]

6.4.3 Member Data Documentation

6.4.3.1 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.

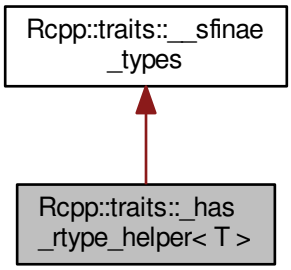
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/matrix_interface.h

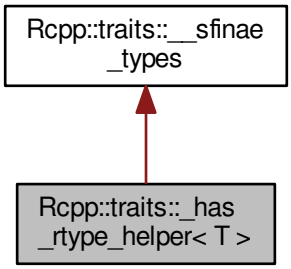
6.5 Rcpp::traits::_has_rtype_helper< T > Class Template Reference

`#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 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]

6.5.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_has_rtype_helper< T >::__test (...) [static], [private]

6.5.3 Member Data Documentation

6.5.3.1 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.

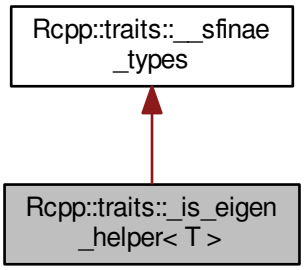
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

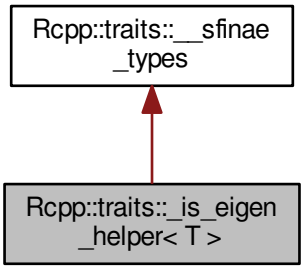
6.6 Rcpp::traits::_is_eigen_helper< T > Class Template Reference

`#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 template<typename T > template<typename U > static __one Rcpp::traits::_is_eigen_helper< T >::__test (_Wrap_type< typename U::StorageKind > *) [static], [private]

6.6.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_is_eigen_helper< T >::__test (...) [static], [private]

6.6.3 Member Data Documentation

6.6.3.1 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.

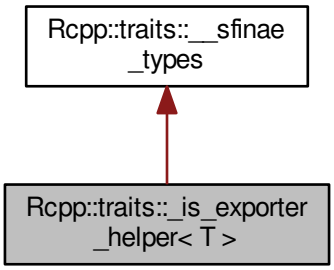
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/is_eigen_base.h

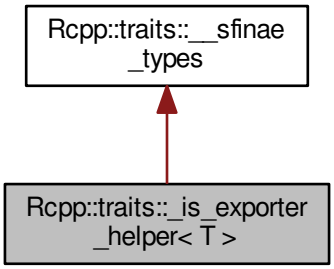
6.7 Rcpp::traits::_is_exporter_helper< T > Class Template Reference

`#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 template<typename T > template<typename U > static __one Rcpp::traits::_is_exporter_helper< T >::__test (_Wrap_type< typename U::r_export_type > *) [static], [private]

6.7.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_is_exporter_helper< T >::__test (...) [static], [private]

6.7.3 Member Data Documentation

6.7.3.1 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.

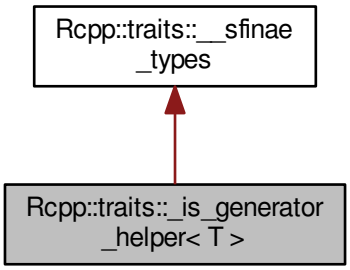
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

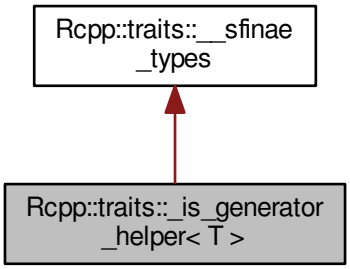
6.8 Rcpp::traits::_is_generator_helper< T > Class Template Reference

`#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 template<typename T > template<typename U > static __one Rcpp::traits::_is_generator_helper< T >::__test (_Wrap_type< typename U::r_generator > *) [static], [private]

6.8.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_is_generator_helper< T >::__test (...) [static], [private]

6.8.3 Member Data Documentation

6.8.3.1 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.

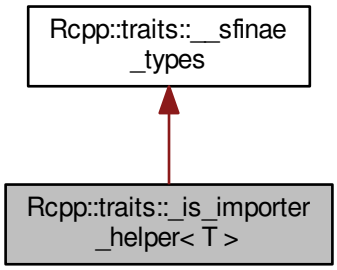
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

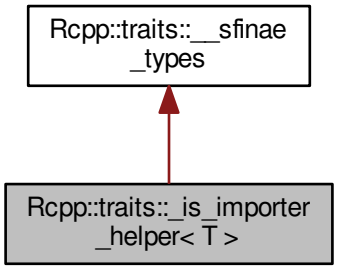
6.9 Rcpp::traits::_is_importer_helper< T > Class Template Reference

`#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 template<typename T > template<typename U > static __one Rcpp::traits::_is_importer_helper< T >::__test (_Wrap_type< typename U::r_import_type > *) [static], [private]

6.9.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_is_importer_helper< T >::__test (...) [static], [private]

6.9.3 Member Data Documentation

6.9.3.1 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.

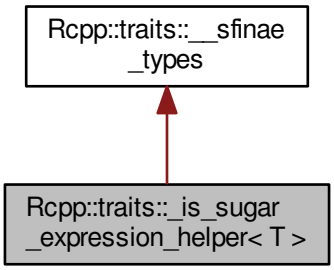
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

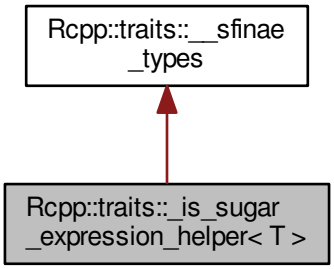
6.10 Rcpp::traits::_is_sugar_expression_helper< T > Class Template Reference

`#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 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]

6.10.2.2 template<typename T > template<typename U > static __two Rcpp::traits::_is_sugar_expression_helper< T >::__test (...) [static], [private]

6.10.3 Member Data Documentation

6.10.3.1 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::_is_importer_helper< T >::_Wrap_type< U > Struct Template Reference

6.12.1 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

6.13 Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U > Struct Template Reference

6.13.1 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

6.14 Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U > Struct Template Reference

6.14.1 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

6.15 Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U > Struct Template Reference

6.15.1 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

6.16 Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U > Struct Template Reference

6.16.1 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

6.17 Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U > Struct Template Reference

6.17.1 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

6.18 Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U > Struct Template Reference

6.18.1 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.

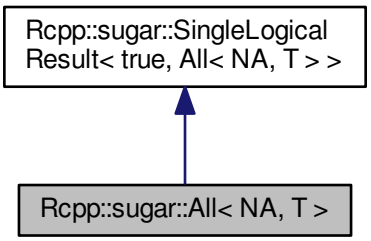
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

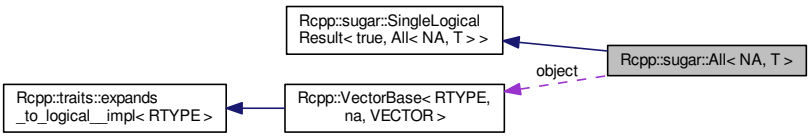
6.19 Rcpp::sugar::All< NA, T > Class Template Reference

`#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 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 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 template<bool NA, typename T > Rcpp::sugar::All< NA, T >::All (const VEC_TYPE & *t*) [inline]

Definition at line 33 of file all.h.

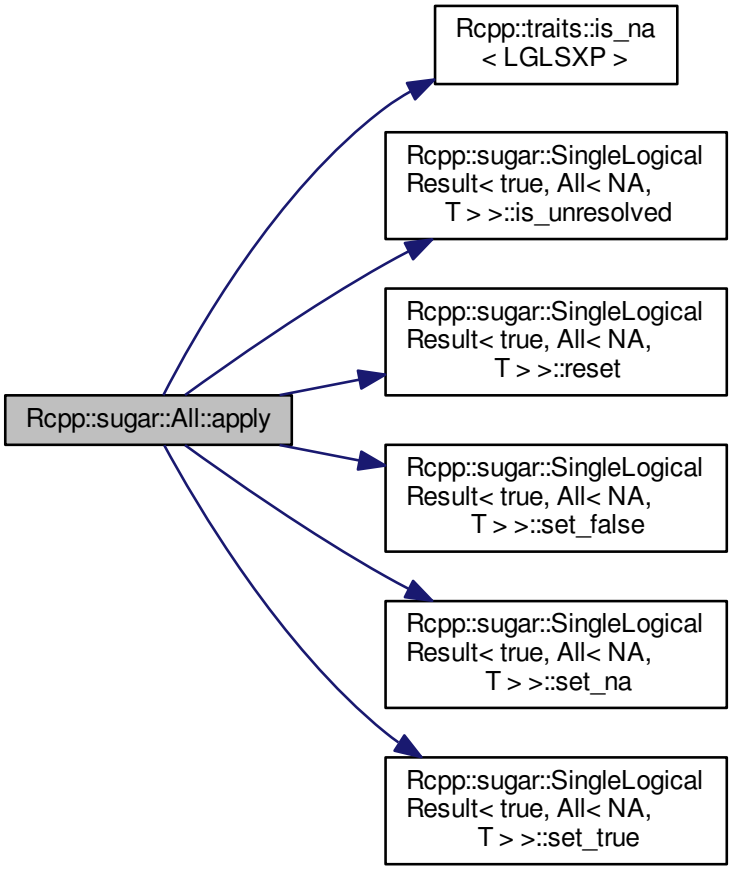
6.19.4 Member Function Documentation

6.19.4.1 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 template<bool NA, typename T > const VEC_TYPE& Rcpp::sugar::All< NA, T >::object [private]

Definition at line 54 of file all.h.

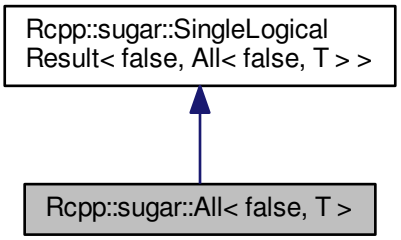
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/all.h

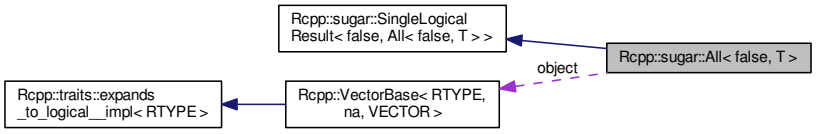
6.20 Rcpp::sugar::All< false, T > Class Template Reference

`#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 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 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 template<typename T > Rcpp::sugar::All< false, T >::All (const VEC_TYPE & *t*) [inline]

Definition at line 64 of file all.h.

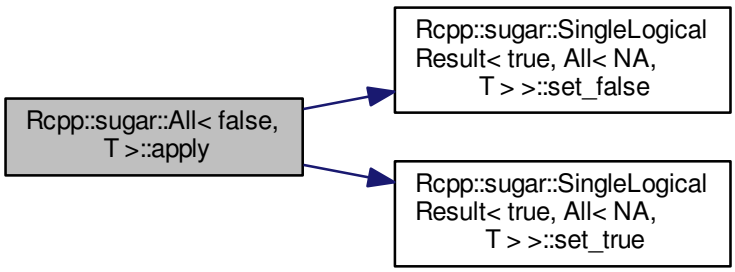
6.20.4 Member Function Documentation

6.20.4.1 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::SingleLogical↩ Result< true, All< NA, T > >::set_true().

Here is the call graph for this function:



6.20.5 Member Data Documentation

6.20.5.1 template<typename T > const VEC_TYPE& Rcpp::sugar::All< false, T >::object [private]

Definition at line 77 of file all.h.

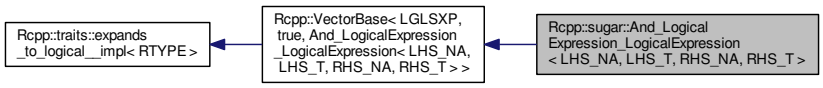
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/all.h

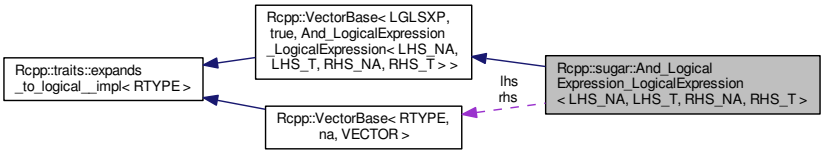
6.21 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

#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

6.21.1 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.

6.21.2 Member Typedef Documentation

6.21.2.1 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.

6.21.2.2 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.

6.21.3 Constructor & Destructor Documentation

6.21.3.1 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.

6.21.4 Member Function Documentation

6.21.4.1 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.

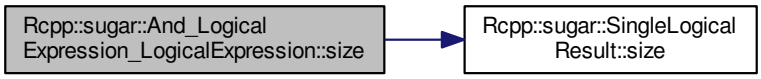
References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.21.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.21.5 Member Data Documentation

6.21.5.1 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.

6.21.5.2 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.

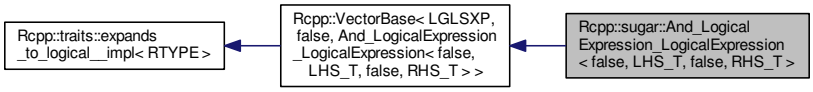
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

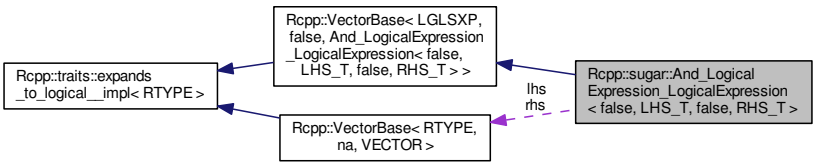
6.22 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.22.1 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.

6.22.2 Member Typedef Documentation

6.22.2.1 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.

6.22.2.2 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.

6.22.3 Constructor & Destructor Documentation

6.22.3.1 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.

6.22.4 Member Function Documentation

6.22.4.1 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.

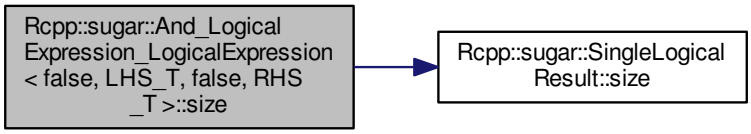
References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.22.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.22.5 Member Data Documentation

6.22.5.1 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.

6.22.5.2 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.

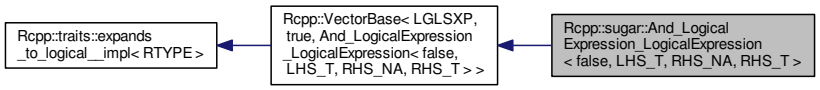
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

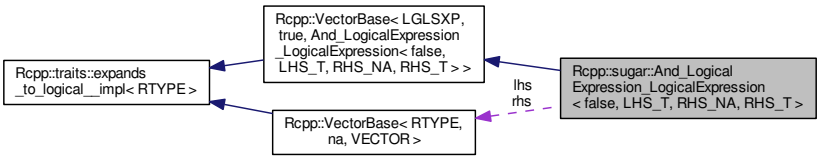
6.23 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.23.1 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.

6.23.2 Member Typedef Documentation

6.23.2.1 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.

6.23.2.2 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.

6.23.3 Constructor & Destructor Documentation

6.23.3.1 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.

6.23.4 Member Function Documentation

6.23.4.1 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.

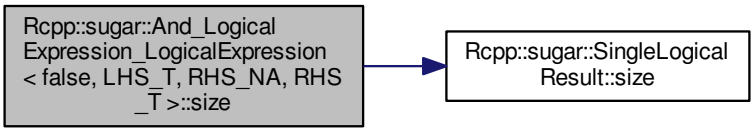
References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.23.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.23.5 Member Data Documentation

6.23.5.1 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.

6.23.5.2 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.

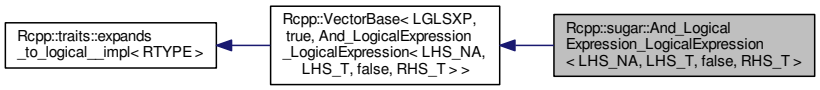
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

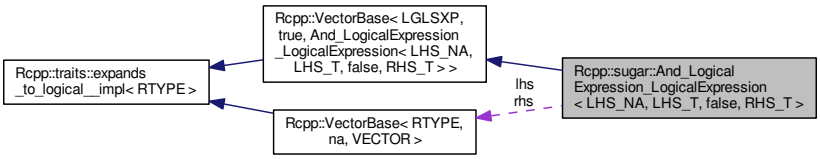
6.24 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.24.1 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.

6.24.2 Member Typedef Documentation

6.24.2.1 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.

6.24.2.2 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.

6.24.3 Constructor & Destructor Documentation

6.24.3.1 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.

6.24.4 Member Function Documentation

6.24.4.1 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.

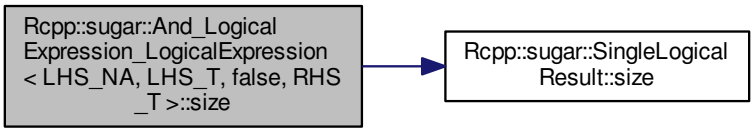
References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.24.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.24.5 Member Data Documentation

6.24.5.1 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.

6.24.5.2 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.

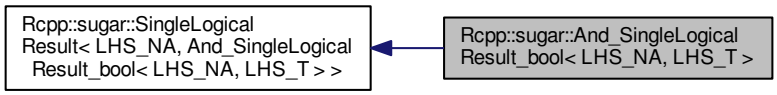
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

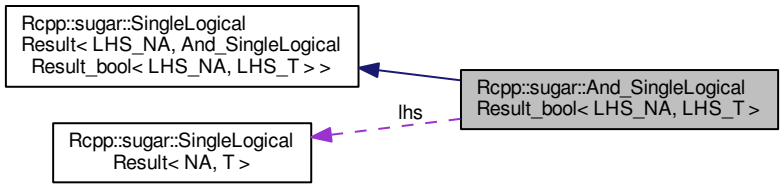
6.25 Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference

`#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

6.25.1 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.

6.25.2 Member Typedef Documentation

6.25.2.1 template<bool LHS_NA, typename LHS_T> typedef SingleLogicalResult< LHS_NA , And_SingleLogical↩ Result_bool<LHS_NA,LHS_T> > Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::BASE

Definition at line 182 of file and.h.

6.25.2.2 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.

6.25.3 Constructor & Destructor Documentation

6.25.3.1 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.

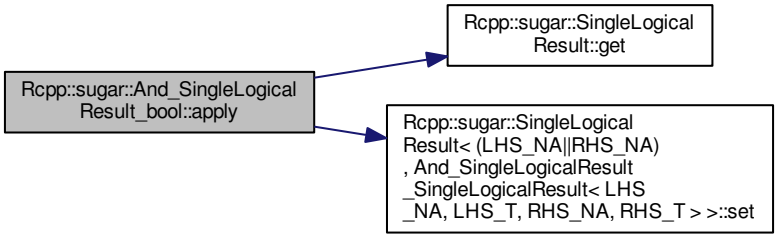
6.25.4 Member Function Documentation

6.25.4.1 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_SingleLogical↩ Result< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< L↩ HS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult< (LHS_NA||RHS_NA), And_Single↩ LogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:



6.25.5 Member Data Documentation

6.25.5.1 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.

6.25.5.2 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.

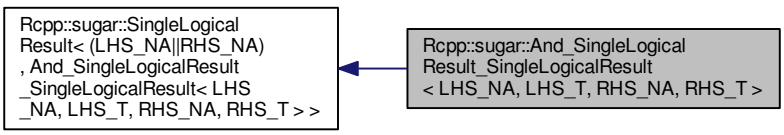
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

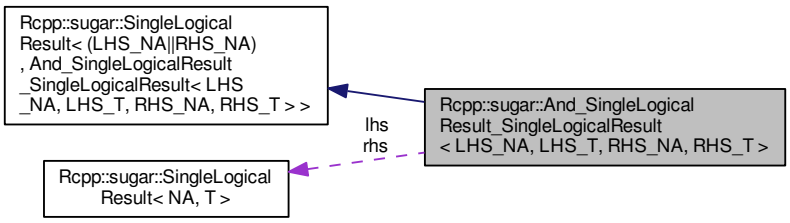
6.26 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.26.1 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.

6.26.2 Member Typedef Documentation

6.26.2.1 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.

6.26.2.2 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.

6.26.2.3 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.

6.26.3 Constructor & Destructor Documentation

6.26.3.1 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_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 43 of file and.h.

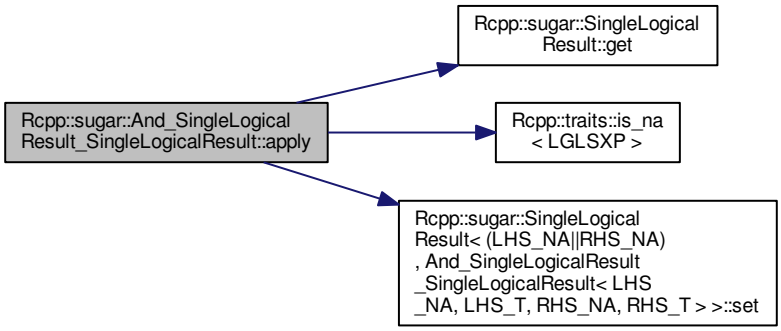
6.26.4 Member Function Documentation

6.26.4.1 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:



6.26.5 Member Data Documentation

6.26.5.1 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(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), operator&(), operator&&(), 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_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size().

6.26.5.2 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(),

Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), operator&(), operator&&(), 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[]().

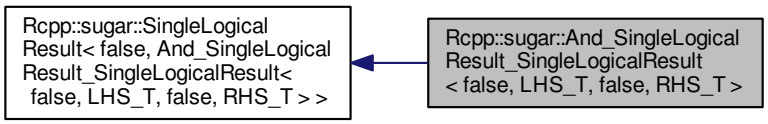
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

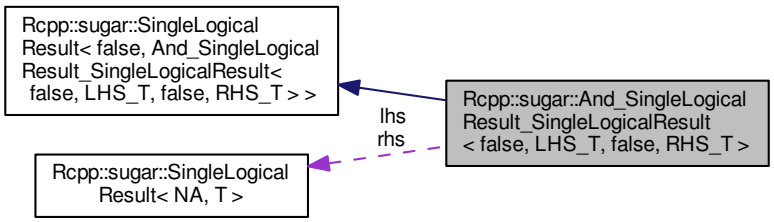
6.27 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.27.1 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.

6.27.2 Member Typedef Documentation

6.27.2.1 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.

6.27.2.2 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.

6.27.2.3 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.

6.27.3 Constructor & Destructor Documentation

6.27.3.1 template<typename LHS_T , typename RHS_T > Rcpp::sugar::And_SingleLogicalResult_SingleLogical↩ Result< false, LHS_T, false, RHS_T >::And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 150 of file and.h.

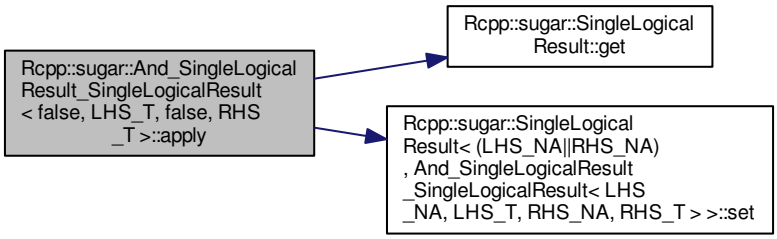
6.27.4 Member Function Documentation

6.27.4.1 template<typename LHS_T , typename RHS_T > void Rcpp::sugar::And_SingleLogicalResult_SingleLogical↩ Result< 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_SingleLogical↩ Result< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< L↩ HS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult< (LHS_NA||RHS_NA), And_Single↩ LogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:



6.27.5 Member Data Documentation

6.27.5.1 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.

6.27.5.2 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.

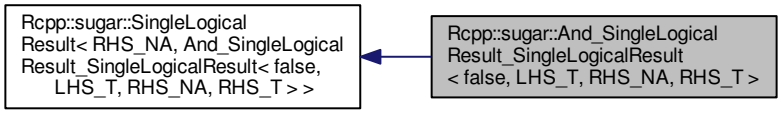
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

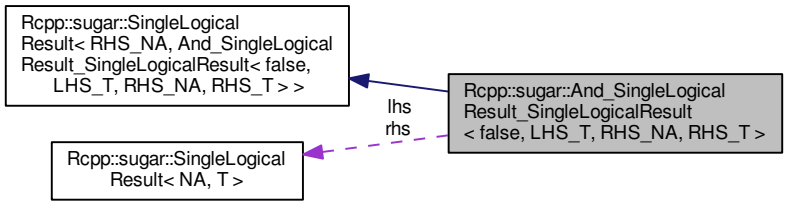
6.28 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.28.1 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.

6.28.2 Member Typedef Documentation

6.28.2.1 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.

6.28.2.2 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.

6.28.2.3 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.

6.28.3 Constructor & Destructor Documentation

6.28.3.1 template<typename LHS_T , bool RHS_NA, typename RHS_T > Rcpp::sugar::And_SingleLogicalResult_↩ SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 115 of file and.h.

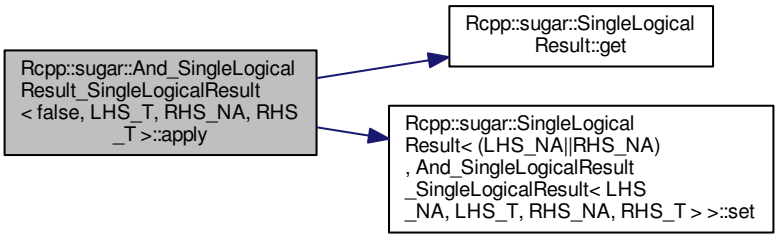
6.28.4 Member Function Documentation

6.28.4.1 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:



6.28.5 Member Data Documentation

6.28.5.1 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.

6.28.5.2 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.

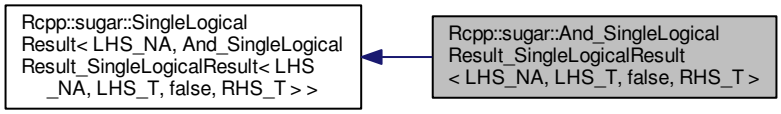
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

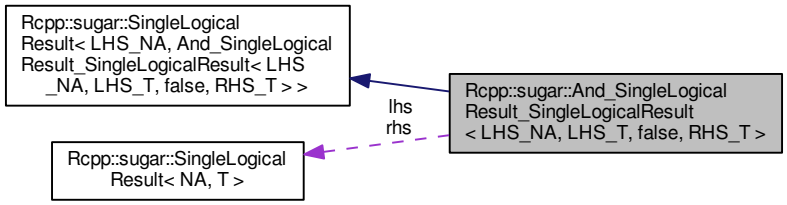
6.29 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.29.1 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.

6.29.2 Member Typedef Documentation

6.29.2.1 template<bool LHS_NA, typename LHS_T , typename RHS_T > typedef SingleLogicalResult< LHS_NA, And_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_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.

6.29.2.2 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.

6.29.2.3 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.

6.29.3 Constructor & Destructor Documentation

6.29.3.1 template<bool LHS_NA, typename LHS_T , typename RHS_T > Rcpp::sugar::And_SingleLogicalResult_Single↩ LogicalResult< LHS_NA, LHS_T, false, RHS_T >::And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 79 of file and.h.

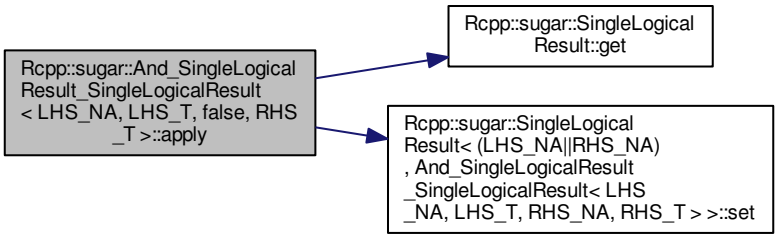
6.29.4 Member Function Documentation

6.29.4.1 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:



6.29.5 Member Data Documentation

6.29.5.1 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.

6.29.5.2 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.

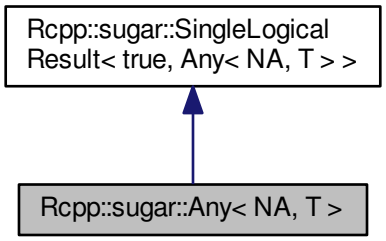
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

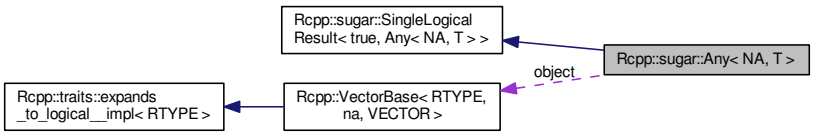
6.30 Rcpp::sugar::Any< NA, T > Class Template Reference

`#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

6.30.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Any< NA, T >

Definition at line 29 of file any.h.

6.30.2 Member Typedef Documentation

6.30.2.1 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.

6.30.2.2 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.

6.30.3 Constructor & Destructor Documentation

6.30.3.1 template<bool NA, typename T > Rcpp::sugar::Any< NA, T >::Any (const VEC_TYPE & *t*) [inline]

Definition at line 33 of file any.h.

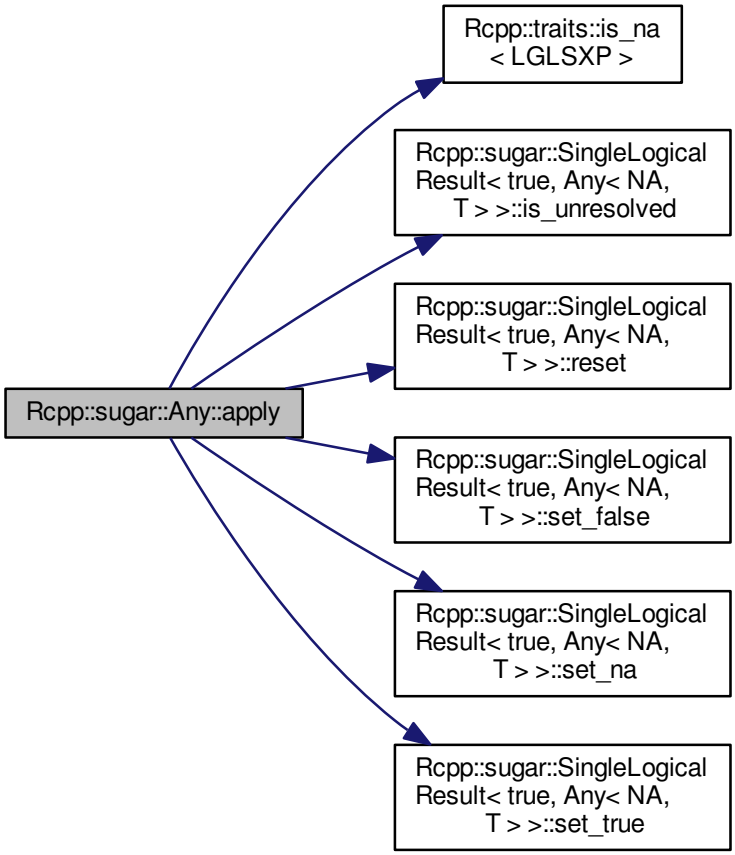
6.30.4 Member Function Documentation

6.30.4.1 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:



6.30.5 Member Data Documentation

6.30.5.1 template<bool NA, typename T > const VEC_TYPE& Rcpp::sugar::Any< NA, T >::object [private]

Definition at line 54 of file any.h.

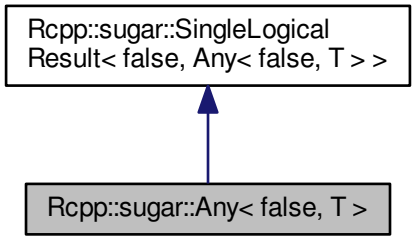
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/any.h

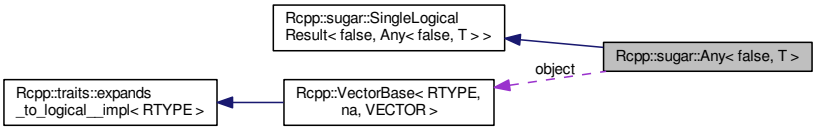
6.31 Rcpp::sugar::Any< false, T > Class Template Reference

`#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

6.31.1 Detailed Description

template<typename T>
class Rcpp::sugar::Any< false, T >

Definition at line 58 of file any.h.

6.31.2 Member Typedef Documentation

6.31.2.1 template<typename T > typedef SingleLogicalResult< false , Any<false,T> > Rcpp::sugar::Any< false, T >::PARENT

Definition at line 61 of file any.h.

6.31.2.2 template<typename T > typedef Rcpp::VectorBase<LGLSXP,false,T> Rcpp::sugar::Any< false, T >::VEC_TYPE

Definition at line 60 of file any.h.

6.31.3 Constructor & Destructor Documentation

6.31.3.1 template<typename T > Rcpp::sugar::Any< false, T >::Any (const VEC_TYPE & *t*) [inline]

Definition at line 62 of file any.h.

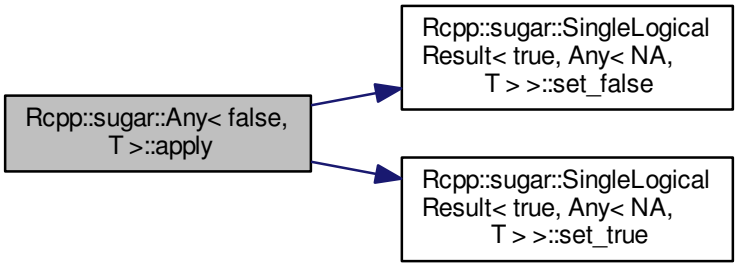
6.31.4 Member Function Documentation

6.31.4.1 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::SingleLogical↩Result< true, Any< NA, T > >::set_true().

Here is the call graph for this function:



6.31.5 Member Data Documentation

6.31.5.1 template<typename T > const VEC_TYPE& Rcpp::sugar::Any< false, T >::object [private]

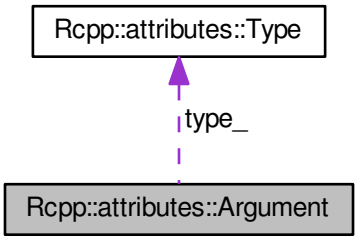
Definition at line 75 of file any.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/any.h

6.32 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_

6.32.1 Detailed Description

Definition at line 203 of file attributes.cpp.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 Rcpp::attributes::Argument::Argument ( ) [inline]

Definition at line 205 of file attributes.cpp.

6.32.2.2 Rcpp::attributes::Argument::Argument (const std::string & *name,* const Type & *type,* const std::string & *defaultValue*) [inline]

Definition at line 206 of file attributes.cpp.

6.32.3 Member Function Documentation

6.32.3.1 const std::string& Rcpp::attributes::Argument::defaultValue ( ) const [inline]

Definition at line 228 of file attributes.cpp.

Referenced by Rcpp::attributes::generateRArgList(), and Rcpp::attributes::printArgument().

6.32.3.2 bool Rcpp::attributes::Argument::empty ( ) const [inline]

Definition at line 213 of file attributes.cpp.

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

6.32.3.3 const std::string& Rcpp::attributes::Argument::name ( ) const [inline]

Definition at line 226 of file attributes.cpp.

Referenced by Rcpp::attributes::generateCpp(), Rcpp::attributes::generateRArgList(), and Rcpp::attributes::printArgument().

6.32.3.4 bool Rcpp::attributes::Argument::operator!= (const Argument & *other*) const [inline]

Definition at line 221 of file attributes.cpp.

6.32.3.5 bool Rcpp::attributes::Argument::operator== (const Argument & *other*) const [inline]

Definition at line 215 of file attributes.cpp.

References defaultValue_, name_, and type_.

6.32.3.6 const Type& Rcpp::attributes::Argument::type ( ) const [inline]

Definition at line 227 of file attributes.cpp.

Referenced by Rcpp::attributes::generateCpp(), Rcpp::attributes::generateRArgList(), and Rcpp::attributes::print↩Argument().

6.32.4 Member Data Documentation

6.32.4.1 std::string Rcpp::attributes::Argument::defaultValue_ [private]

Definition at line 233 of file attributes.cpp.

Referenced by operator==().

6.32.4.2 std::string Rcpp::attributes::Argument::name_ [private]

Definition at line 231 of file attributes.cpp.

Referenced by operator==().

6.32.4.3 Type Rcpp::attributes::Argument::type_ [private]

Definition at line 232 of file attributes.cpp.

Referenced by operator==().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.33 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

6.33.1 Detailed Description

Definition at line 27 of file Named.h.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 Rcpp::Argument::Argument ( ) [inline]

Definition at line 29 of file Named.h.

Referenced by Rcpp::Named(), Rcpp::internal::NamedPlaceHolder::operator()(), and Rcpp::internal::NamedPlaceHolder::operator[]().

6.33.2.2 Rcpp::Argument::Argument (const std::string & *name_*) [inline]

Definition at line 30 of file Named.h.

6.33.3 Member Function Documentation

6.33.3.1 template<typename T > traits::named_object<T> Rcpp::Argument::operator= (const T & *t*) [inline]

Definition at line 33 of file Named.h.

References name.

6.33.4 Member Data Documentation

6.33.4.1 std::string Rcpp::Argument::name

Definition at line 37 of file Named.h.

Referenced by Rcpp::Named(), and operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Named.h

6.34 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

6.34.1 Detailed Description

template<typename T>
class Rcpp::Armor< T >

Definition at line 24 of file Armor.h.

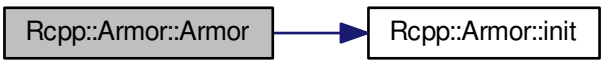
6.34.2 Constructor & Destructor Documentation

6.34.2.1 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:

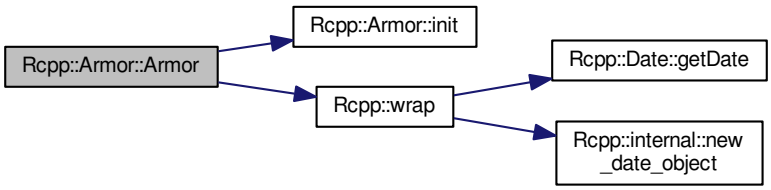


6.34.2.2 template<typename T > template<typename U > Rcpp::Armor< T >::Armor (U *x*)

Definition at line 25 of file protection.h.

References Rcpp::Armor< T >::init(), and Rcpp::wrap().

Here is the call graph for this function:



6.34.2.3 template<typename T> Rcpp::Armor< T >::~Armor ( ) [inline]

Definition at line 40 of file Armor.h.

6.34.2.4 template<typename T> Rcpp::Armor< T >::Armor (const Armor< T > &) [private]

6.34.3 Member Function Documentation

6.34.3.1 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.

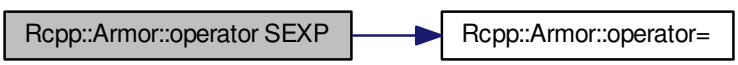
Referenced by Rcpp::Armor< T >::Armor().

6.34.3.2 template<typename T> Rcpp::Armor< T >::operator SEXP ( ) const [inline]

Definition at line 33 of file Armor.h.

References Rcpp::Armor< T >::data, and Rcpp::Armor< T >::operator=().

Here is the call graph for this function:

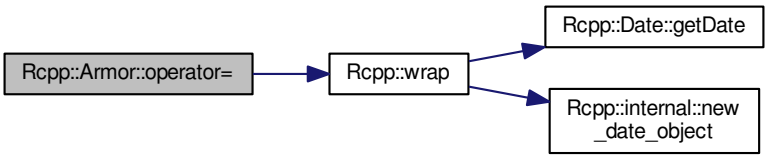


6.34.3.3 template<typename T> template<typename U > Armor<T>& Rcpp::Armor< T >::operator= (const U & *x*) [inline]

Definition at line 31 of file protection.h.

References Rcpp::Armor< T >::data, Rcpp::Armor< T >::index, and Rcpp::wrap().

Here is the call graph for this function:



6.34.3.4 template<typename T> template<typename U > Armor& Rcpp::Armor< T >::operator= (const U & *x*) [inline]

Referenced by Rcpp::Armor< T >::operator SEXP().

6.34.3.5 template<typename T> Armor& Rcpp::Armor< T >::operator= (const Armor< T > &) [private]

6.34.4 Member Data Documentation

6.34.4.1 template<typename T> SEXP Rcpp::Armor< T >::data [private]

Definition at line 49 of file Armor.h.

Referenced by Rcpp::Armor< T >::init(), Rcpp::Armor< T >::operator SEXP(), and Rcpp::Armor< T >::operator=().

6.34.4.2 template<typename T> PROTECT_INDEX Rcpp::Armor< T >::index [private]

Definition at line 50 of file Armor.h.

Referenced by Rcpp::Armor< T >::init(), and Rcpp::Armor< T >::operator=().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/protection/Armor.h
- inst/include/Rcpp/api/meat/protection.h

6.35 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
- const std::vector< std::string > & roxygen () const

Private Attributes

- std::string name_
- std::vector< Param > params_
- Function function_
- std::vector< std::string > roxygen_

6.35.1 Detailed Description

Definition at line 306 of file attributes.cpp.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 Rcpp::attributes::Attribute::Attribute ( ) [inline]

Definition at line 308 of file attributes.cpp.

6.35.2.2 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 309 of file attributes.cpp.

6.35.3 Member Function Documentation

6.35.3.1 bool Rcpp::attributes::Attribute::empty ( ) const [inline]

Definition at line 317 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.35.3.2 std::string Rcpp::attributes::Attribute::exportedCppName ( ) const [inline]

Definition at line 365 of file attributes.cpp.

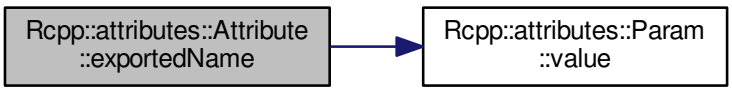
6.35.3.3 std::string Rcpp::attributes::Attribute::exportedName ( ) const [inline]

Definition at line 347 of file attributes.cpp.

References Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::isRoxygenCpp(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



6.35.3.4 const Function& Rcpp::attributes::Attribute::function ( ) const [inline]

Definition at line 341 of file attributes.cpp.

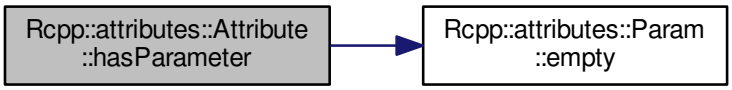
Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::generateCpp(), Rcpp↩
::attributes::isRoxygenCpp(), Rcpp::attributes::operator<<(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

6.35.3.5 bool Rcpp::attributes::Attribute::hasParameter (const std::string & *name*) const [inline]

Definition at line 337 of file attributes.cpp.

References Rcpp::attributes::Param::empty().

Here is the call graph for this function:



6.35.3.6 bool Rcpp::attributes::Attribute::isExportedFunction ( ) const [inline]

Definition at line 343 of file attributes.cpp.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::generateCpp(), and Rcpp↩
::attributes::isRoxygenCpp().

6.35.3.7 const std::string& Rcpp::attributes::Attribute::name ( ) const [inline]

Definition at line 331 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.35.3.8 bool Rcpp::attributes::Attribute::operator!= (const Attribute & *other*) const [inline]

Definition at line 326 of file attributes.cpp.

6.35.3.9 bool Rcpp::attributes::Attribute::operator== (const Attribute & *other*) const [inline]

Definition at line 319 of file attributes.cpp.

References function_, name_, params_, and roxygen_.

6.35.3.10 Param Rcpp::attributes::Attribute::paramNamed (const std::string & *name*) const

Definition at line 1014 of file attributes.cpp.

6.35.3.11 const std::vector<Param>& Rcpp::attributes::Attribute::params ( ) const [inline]

Definition at line 333 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

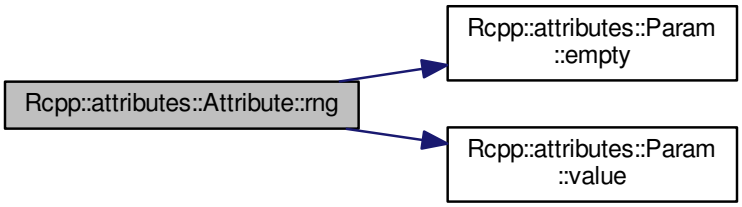
6.35.3.12 bool Rcpp::attributes::Attribute::rng ( ) const [inline]

Definition at line 371 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::kParamValueTRUE, and Rcpp::attributes::Param::value().

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

Here is the call graph for this function:



6.35.3.13 const std::vector<std::string>& Rcpp::attributes::Attribute::roxygen ( ) const [inline]

Definition at line 380 of file attributes.cpp.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

6.35.4 Member Data Documentation

6.35.4.1 Function Rcpp::attributes::Attribute::function_ [private]

Definition at line 385 of file attributes.cpp.

Referenced by operator==().

6.35.4.2 std::string Rcpp::attributes::Attribute::name_ [private]

Definition at line 383 of file attributes.cpp.

Referenced by operator==().

6.35.4.3 std::vector<Param> Rcpp::attributes::Attribute::params_ [private]

Definition at line 384 of file attributes.cpp.

Referenced by operator==().

6.35.4.4 std::vector<std::string> Rcpp::attributes::Attribute::roxygen_ [private]

Definition at line 386 of file attributes.cpp.

Referenced by operator==().

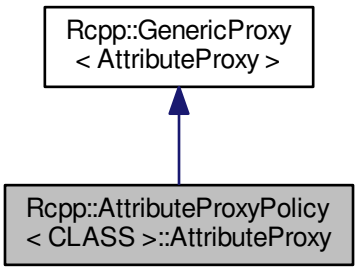
The documentation for this class was generated from the following file:

- src/attributes.cpp

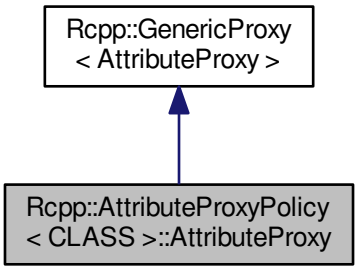
6.36 Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy Class Reference

`#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

6.36.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy

Definition at line 27 of file AttributeProxy.h.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 template<typename CLASS> Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::AttributeProxy (CLASS & *v*, const std::string & *name*) [inline]

Definition at line 29 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< Vector< RTYPE, StoragePolicy > >::attr().

6.36.3 Member Function Documentation

6.36.3.1 template<typename CLASS> SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get ( ) const [inline], [private]

Definition at line 48 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=().

6.36.3.2 template<typename CLASS > Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator SEXP ( ) const [inline]

Definition at line 44 of file proxy.h.

6.36.3.3 template<typename CLASS > template<typename T > Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator T ( ) const

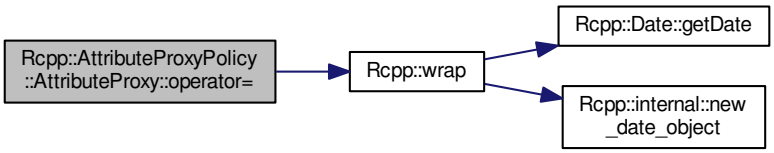
Definition at line 39 of file proxy.h.

6.36.3.4 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::wrap().

Here is the call graph for this function:

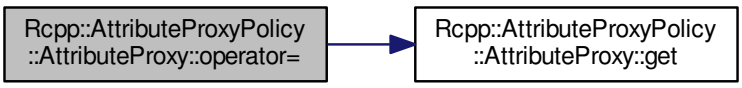


6.36.3.5 template<typename CLASS> AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (const AttributeProxy & *rhs*) [inline]

Definition at line 33 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get().

Here is the call graph for this function:



6.36.3.6 template<typename CLASS> template<typename T > AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (const T & *rhs*)

6.36.3.7 template<typename CLASS> void Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set (SEXP *x*) [inline], [private]

Definition at line 51 of file AttributeProxy.h.

6.36.4 Member Data Documentation

6.36.4.1 template<typename CLASS> SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr_name [private]

Definition at line 46 of file AttributeProxy.h.

6.36.4.2 template<typename CLASS> CLASS& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent [private]

Definition at line 45 of file AttributeProxy.h.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/AttributeProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.37 Rcpp::AttributeProxyPolicy< CLASS > Class Template Reference

`#include <AttributeProxy.h>`

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

6.37.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >

Definition at line 24 of file AttributeProxy.h.

6.37.2 Member Function Documentation

6.37.2.1 template<typename CLASS> AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (const std::string & *name*) [inline]

Definition at line 73 of file AttributeProxy.h.

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), Rcpp::sugar::na_omit_impl(), Rcpp::Timer::operator SEXP(), Rcpp::internal::range_wrap_dispatch___impl__pair(), stack_trace(), and Rcpp::Vector< INTSXP >::Vector().

6.37.2.2 template<typename CLASS> const_AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (const std::string & *name*) const [inline]

Definition at line 76 of file AttributeProxy.h.

6.37.2.3 template<typename CLASS> std::vector<std::string> Rcpp::AttributeProxyPolicy< CLASS >::attributeNames ( ) const [inline]

Definition at line 80 of file AttributeProxy.h.

6.37.2.4 template<typename CLASS> bool Rcpp::AttributeProxyPolicy< CLASS >::hasAttribute (const std::string & *attr*) const [inline]

Definition at line 90 of file AttributeProxy.h.

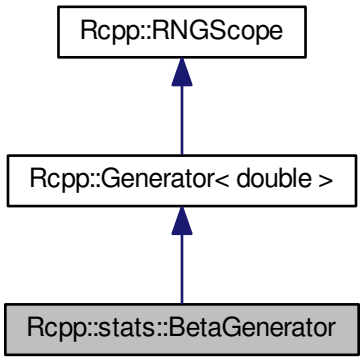
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/AttributeProxy.h

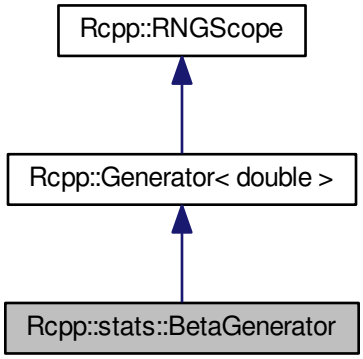
6.38 Rcpp::stats::BetaGenerator Class Reference

```
#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

6.38.1 Detailed Description

Definition at line 28 of file rbeta.h.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 Rcpp::stats::BetaGenerator::BetaGenerator (double *a_*, double *b_*) [inline]

Definition at line 30 of file rbeta.h.

6.38.3 Member Function Documentation

6.38.3.1 double Rcpp::stats::BetaGenerator::operator() ( ) const [inline]

Definition at line 32 of file rbeta.h.

References a, and b.

6.38.4 Member Data Documentation

6.38.4.1 double Rcpp::stats::BetaGenerator::a [private]

Definition at line 36 of file rbeta.h.

Referenced by operator()().

6.38.4.2 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

6.39 Rcpp::traits::is_convertible< T, U >::Big Struct Reference

Public Attributes

- char dummy [2]

6.39.1 Detailed Description

template<typename T, typename U>
struct Rcpp::traits::is_convertible< T, U >::Big

Definition at line 32 of file is_convertible.h.

6.39.2 Member Data Documentation

6.39.2.1 template<typename T , typename U > char Rcpp::traits::is_convertible< T, U >::Big::dummy[2]

Definition at line 32 of file is_convertible.h.

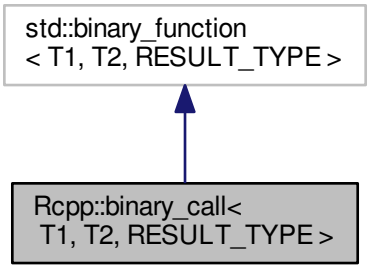
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_convertible.h

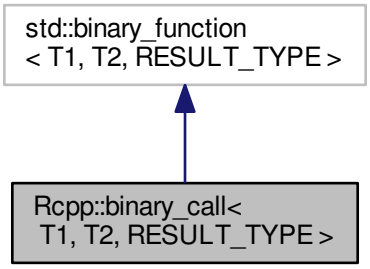
6.40 Rcpp::binary_call< T1, T2, RESULT_TYPE > Class Template Reference

`#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

6.40.1 Detailed Description

template<typename T1, typename T2, typename RESULT_TYPE = SEXP>
class Rcpp::binary_call< T1, T2, RESULT_TYPE >

Definition at line 197 of file Language.h.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP> Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (Language *call_*) [inline]

Definition at line 199 of file Language.h.

6.40.2.2 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 200 of file Language.h.

6.40.2.3 template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP> Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (Function *fun*) [inline]

Definition at line 201 of file Language.h.

6.40.3 Member Function Documentation

6.40.3.1 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 203 of file Language.h.

References Rcpp::fixed_call< RESULT_TYPE >::call.

6.40.4 Member Data Documentation

6.40.4.1 template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP> Language Rcpp::binary_call< T1, T2, RESULT_TYPE >::call [private]

Definition at line 210 of file Language.h.

6.40.4.2 template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP> Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy1 [private]

Definition at line 211 of file Language.h.

6.40.4.3 template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP> Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy2 [private]

Definition at line 212 of file Language.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.41 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

6.41.1 Detailed Description

template<int RTYPE, typename E>
class Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >

Definition at line 52 of file cbind.h.

6.41.2 Member Typedef Documentation

6.41.2.1 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.

6.41.3 Member Function Documentation

6.41.3.1 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.

6.41.3.2 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.

6.41.3.3 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.

6.41.3.4 template<int RTYPE, typename E> Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator E & ( ) [inline]

Definition at line 76 of file cbind.h.

6.41.3.5 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.

6.41.3.6 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.

6.41.3.7 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.

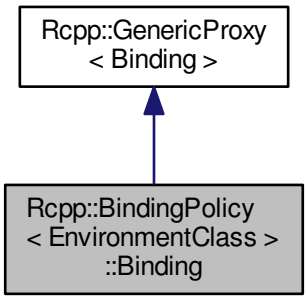
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

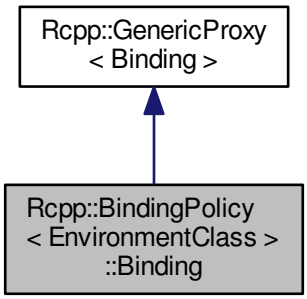
6.42 Rcpp::BindingPolicy< EnvironmentClass >::Binding Class Reference

`#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

6.42.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::Binding

Definition at line 27 of file Binding.h.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 template<typename EnvironmentClass > Rcpp::BindingPolicy< EnvironmentClass >::Binding::Binding (EnvironmentClass & *env_*, const std::string & *name_*) [inline]

Definition at line 29 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::operator[]().

6.42.3 Member Function Documentation

6.42.3.1 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.

6.42.3.2 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.

6.42.3.3 template<typename EnvironmentClass > SEXP 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=().

6.42.3.4 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.

6.42.3.5 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.

6.42.3.6 template<typename CLASS > template<typename T > Rcpp::BindingPolicy< CLASS >::Binding::operator T ( ) const

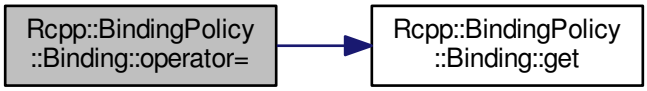
Definition at line 137 of file proxy.h.

6.42.3.7 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().

Here is the call graph for this function:



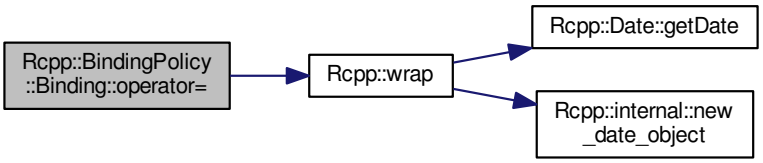
6.42.3.8 template<typename EnvironmentClass > template<typename WRAPPABLE > Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (const WRAPPABLE & *rhs*)

6.42.3.9 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:



6.42.3.10 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.

6.42.3.11 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.

6.42.4 Member Data Documentation

6.42.4.1 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< Environment↩ Class >::const_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::↩ BindingPolicy< EnvironmentClass >::const_Binding::get(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::const_↩ Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< Environment↩ Class >::Binding::unlock().

6.42.4.2 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< Environment↩ Class >::const_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::↩ BindingPolicy< EnvironmentClass >::const_Binding::get(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::const_↩ Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< Environment↩ Class >::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

6.43 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)

6.43.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >

Definition at line 24 of file Binding.h.

6.43.2 Member Function Documentation

6.43.2.1 template<typename EnvironmentClass > const_Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[ ] (const std::string & *name*) const [inline]

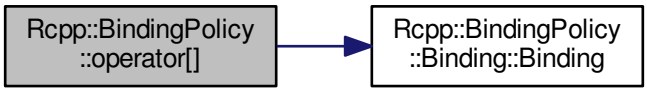
Definition at line 98 of file Binding.h.

6.43.2.2 template<typename EnvironmentClass > Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[ ] (const std::string & *name*) [inline]

Definition at line 101 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::Binding().

Here is the call graph for this function:



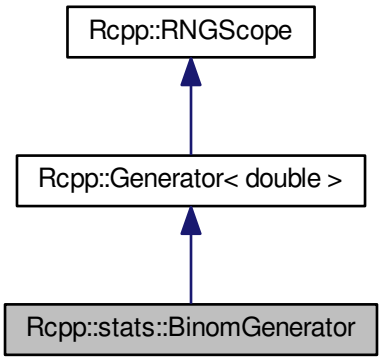
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/Binding.h

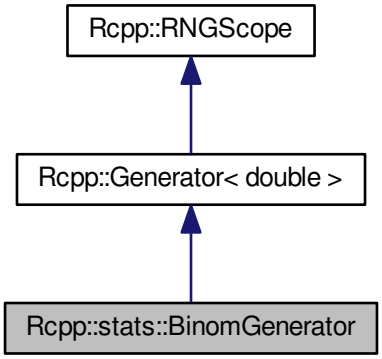
6.44 Rcpp::stats::BinomGenerator Class Reference

`#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

6.44.1 Detailed Description

Definition at line 28 of file rbinom.h.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 Rcpp::stats::BinomGenerator::BinomGenerator (double *nin_*, double *pp_*) [inline]

Definition at line 30 of file rbinom.h.

6.44.3 Member Function Documentation

6.44.3.1 double Rcpp::stats::BinomGenerator::operator() ( ) const [inline]

Definition at line 31 of file rbinom.h.

References nin, and pp.

6.44.4 Member Data Documentation

6.44.4.1 double Rcpp::stats::BinomGenerator::nin [private]

Definition at line 35 of file rbinom.h.

Referenced by operator()().

6.44.4.2 double Rcpp::stats::BinomGenerator::pp [private]

Definition at line 35 of file rbinom.h.

Referenced by operator()().

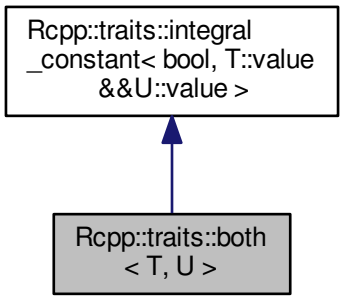
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rbinom.h

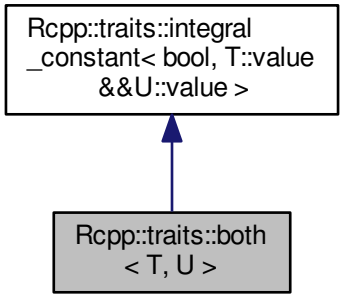
6.45 Rcpp::traits::both< T, U > Struct Template Reference

`#include <integral_constant.h>`

Inheritance diagram for Rcpp::traits::both< T, U >:



Collaboration diagram for Rcpp::traits::both< T, U >:



Additional Inherited Members

6.45.1 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

6.46 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

6.46.1 Detailed Description

Definition at line 2 of file convolve10_cpp.h.

6.46.2 Member Typedef Documentation

6.46.2.1 typedef double* Cache::iterator

Definition at line 5 of file convolve10_cpp.h.

6.46.2.2 typedef double* Cache::iterator

Definition at line 18 of file convolve9_cpp.cpp.

6.46.2.3 typedef double& Cache::proxy

Definition at line 4 of file convolve10_cpp.h.

6.46.2.4 typedef double& Cache::proxy

Definition at line 17 of file convolve9_cpp.cpp.

6.46.3 Constructor & Destructor Documentation

6.46.3.1 Cache::Cache (iterator *data_*) [inline]

Definition at line 7 of file convolve10_cpp.h.

6.46.3.2 Cache::Cache (iterator *data_*) [inline]

Definition at line 20 of file convolve9_cpp.cpp.

6.46.4 Member Function Documentation

6.46.4.1 proxy Cache::ref (int *i*) [inline]

Definition at line 9 of file convolve10_cpp.h.

References data.

6.46.4.2 proxy Cache::ref (int *i*) const [inline]

Definition at line 10 of file convolve10_cpp.h.

References data.

6.46.4.3 proxy Cache::ref (int *i*) [inline]

Definition at line 22 of file convolve9_cpp.cpp.

References data.

6.46.4.4 proxy Cache::ref (int *i*) const [inline]

Definition at line 23 of file convolve9_cpp.cpp.

References data.

6.46.5 Member Data Documentation

6.46.5.1 iterator Cache::data [private]

Definition at line 13 of file convolve10_cpp.h.

Referenced by ref().

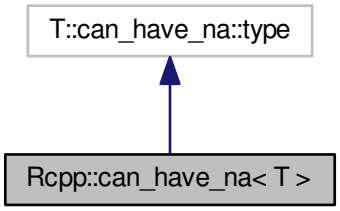
The documentation for this class was generated from the following files:

- inst/examples/ConvolveBenchmarks/convolve10_cpp.h
- inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp

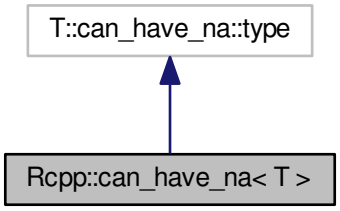
6.47 Rcpp::can_have_na< T > Struct Template Reference

`#include <can_have_na.h>`

Inheritance diagram for Rcpp::can_have_na< T >:



Collaboration diagram for Rcpp::can_have_na< T >:



6.47.1 Detailed Description

template<typename T>
struct Rcpp::can_have_na< T >

Definition at line 28 of file can_have_na.h.

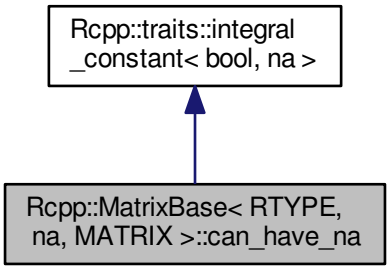
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/can_have_na.h

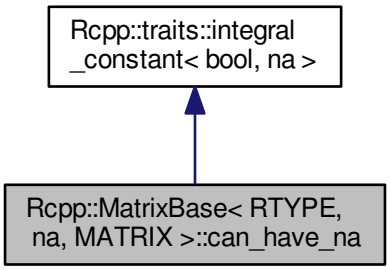
6.48 Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na Struct Reference

`#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

6.48.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na

Definition at line 35 of file MatrixBase.h.

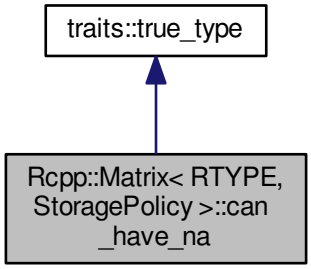
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

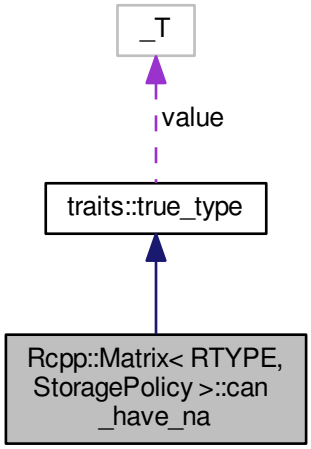
6.49 Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na Struct Reference

`#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

6.49.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na

Definition at line 33 of file Matrix.h.

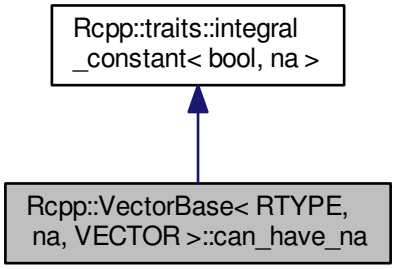
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/Matrix.h

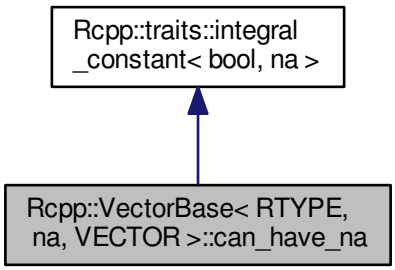
6.50 Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na Struct Reference

`#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

6.50.1 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.

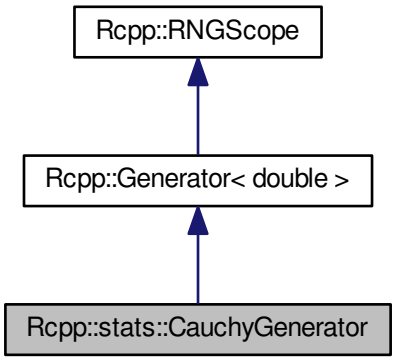
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

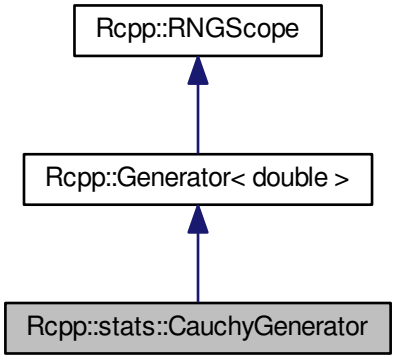
6.51 Rcpp::stats::CauchyGenerator Class Reference

```
#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

6.51.1 Detailed Description

Definition at line 28 of file rcauchy.h.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 Rcpp::stats::CauchyGenerator::CauchyGenerator (double *location_*, double *scale_*) [inline]

Definition at line 31 of file rcauchy.h.

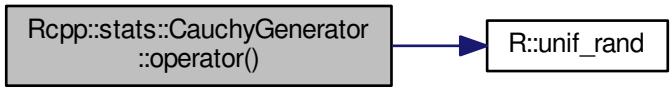
6.51.3 Member Function Documentation

6.51.3.1 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:



6.51.4 Member Data Documentation

6.51.4.1 double Rcpp::stats::CauchyGenerator::location [private]

Definition at line 39 of file rcauchy.h.

Referenced by operator()(), and Rcpp::stats::CauchyGenerator_1::operator()().

6.51.4.2 double Rcpp::stats::CauchyGenerator::scale [private]

Definition at line 39 of file rcauchy.h.

Referenced by operator()().

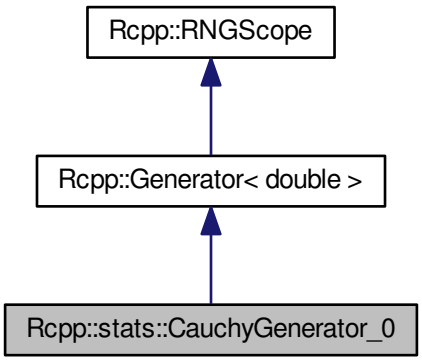
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

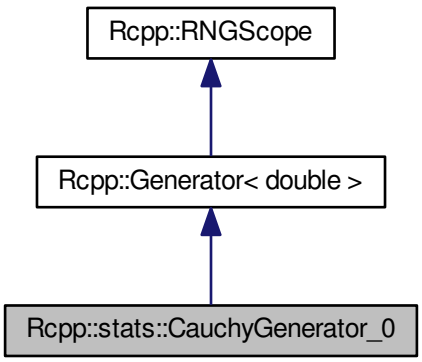
6.52 Rcpp::stats::CauchyGenerator_0 Class Reference

`#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

6.52.1 Detailed Description

Definition at line 56 of file rcauchy.h.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 Rcpp::stats::CauchyGenerator_0::CauchyGenerator_0 ( ) [inline]

Definition at line 59 of file rcauchy.h.

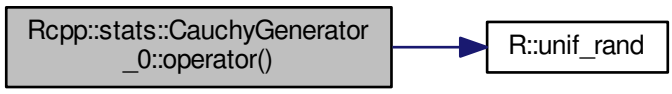
6.52.3 Member Function Documentation

6.52.3.1 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:



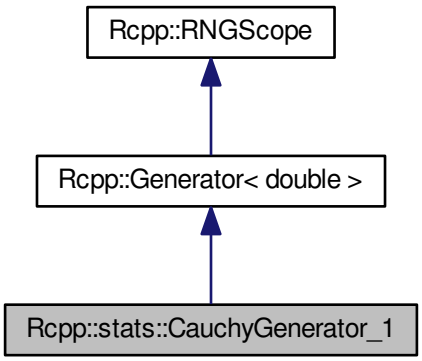
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

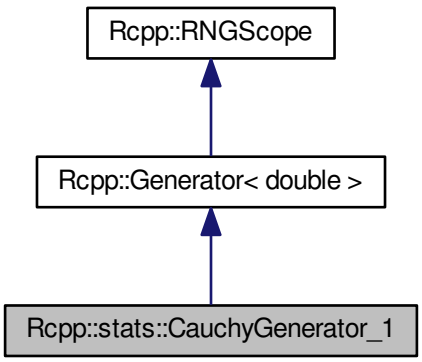
6.53 Rcpp::stats::CauchyGenerator_1 Class Reference

`#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

6.53.1 Detailed Description

Definition at line 42 of file rcauchy.h.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 Rcpp::stats::CauchyGenerator_1::CauchyGenerator_1 (double *location_*) [inline]

Definition at line 45 of file rcauchy.h.

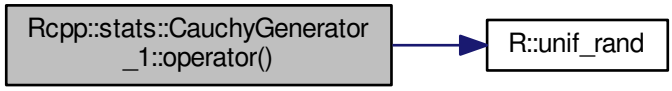
6.53.3 Member Function Documentation

6.53.3.1 double Rcpp::stats::CauchyGenerator_1::operator() ( ) const [inline]

Definition at line 48 of file rcauchy.h.

References Rcpp::stats::CauchyGenerator::location, and R::unif_rand().

Here is the call graph for this function:



6.53.4 Member Data Documentation

6.53.4.1 double Rcpp::stats::CauchyGenerator_1::location [private]

Definition at line 53 of file rcauchy.h.

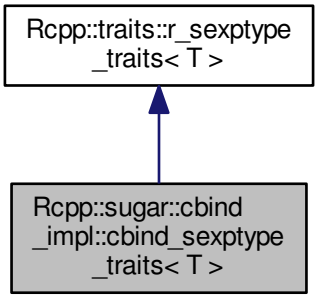
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

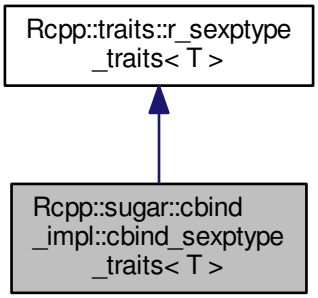
6.54 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T > Struct Template Reference

`#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

6.54.1 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

6.55 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP > Struct Template Reference

`#include <cbind.h>`

Public Types

- enum { rtype = STRSXP }

6.55.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP >

Definition at line 35 of file cbind.h.

6.55.2 Member Enumeration Documentation

6.55.2.1 anonymous enum

Enumerator

rtype

Definition at line 36 of file cbind.h.

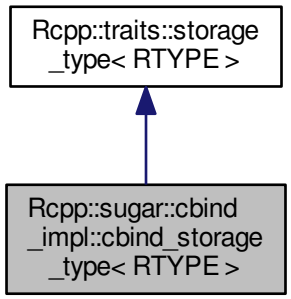
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

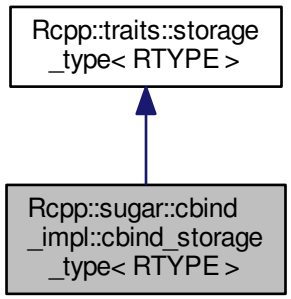
6.56 Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE > Struct Template Reference

`#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

6.56.1 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

6.57 Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP > Struct Template Reference

`#include <cbind.h>`

Public Types

- typedef bool type

6.57.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP >

Definition at line 45 of file cbind.h.

6.57.2 Member Typedef Documentation

6.57.2.1 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

6.58 Rcpp::traits::char_type< T > Struct Template Reference

`#include <char_type.h>`

Public Types

- typedef T::value_type type

6.58.1 Detailed Description

template<typename T>
struct Rcpp::traits::char_type< T >

Definition at line 30 of file char_type.h.

6.58.2 Member Typedef Documentation

6.58.2.1 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

6.59 Rcpp::traits::char_type< const char * > Struct Template Reference

`#include <char_type.h>`

Public Types

- typedef char type

6.59.1 Detailed Description

template<>
struct Rcpp::traits::char_type< const char * >

Definition at line 37 of file char_type.h.

6.59.2 Member Typedef Documentation

6.59.2.1 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

6.60 Rcpp::traits::char_type< const wchar_t * > Struct Template Reference

`#include <char_type.h>`

Public Types

- typedef wchar_t type

6.60.1 Detailed Description

template<>
struct Rcpp::traits::char_type< const wchar_t * >

Definition at line 34 of file char_type.h.

6.60.2 Member Typedef Documentation

6.60.2.1 typedef wchar_t Rcpp::traits::char_type< const wchar_t * >::type

Definition at line 35 of file char_type.h.

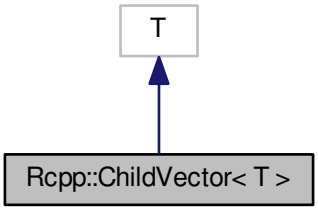
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/char_type.h

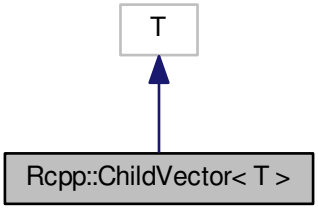
6.61 Rcpp::ChildVector< T > Class Template Reference

`#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

6.61.1 Detailed Description

template<typename T>
class Rcpp::ChildVector< T >

Definition at line 26 of file ChildVector.h.

6.61.2 Constructor & Destructor Documentation

6.61.2.1 template<typename T> Rcpp::ChildVector< T >::ChildVector (SEXP *data_*, SEXP *parent_*, R_xlen_t *i_*) [inline]

Definition at line 30 of file ChildVector.h.

6.61.2.2 template<typename T> Rcpp::ChildVector< T >::ChildVector (const ChildVector< T > & *other*) [inline]

Definition at line 35 of file ChildVector.h.

6.61.3 Member Function Documentation

6.61.3.1 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.

6.61.3.2 template<typename T> ChildVector& Rcpp::ChildVector< T >::operator= (const T & *other*) [inline]

Definition at line 50 of file ChildVector.h.

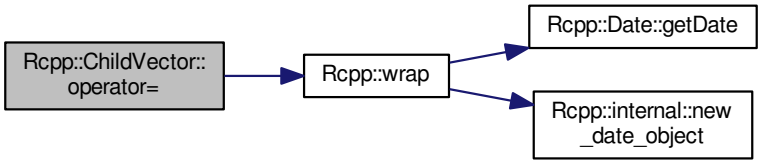
References Rcpp::ChildVector< T >::i, and Rcpp::ChildVector< T >::parent.

6.61.3.3 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:



6.61.4 Member Data Documentation

6.61.4.1 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=().

6.61.4.2 template<typename T> SEXP Rcpp::ChildVector< T >::parent [private]

Definition at line 70 of file ChildVector.h.

Referenced by Rcpp::ChildVector< T >::operator=().

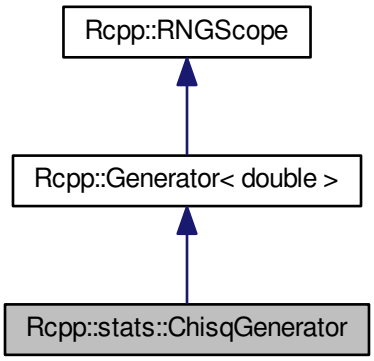
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ChildVector.h

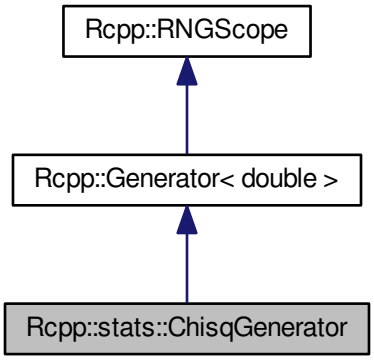
6.62 Rcpp::stats::ChisqGenerator Class Reference

```
#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

6.62.1 Detailed Description

Definition at line 28 of file rchisq.h.

6.62.2 Constructor & Destructor Documentation

6.62.2.1 Rcpp::stats::ChisqGenerator::ChisqGenerator (double *df_*) [inline]

Definition at line 31 of file rchisq.h.

6.62.3 Member Function Documentation

6.62.3.1 double Rcpp::stats::ChisqGenerator::operator() ( ) const [inline]

Definition at line 33 of file rchisq.h.

References df_2.

6.62.4 Member Data Documentation

6.62.4.1 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

6.63 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

6.63.1 Detailed Description

template<int RTYPE, bool NA>
struct Rcpp::sugar::clamp_operator< RTYPE, NA >

Definition at line 29 of file clamp.h.

6.63.2 Member Typedef Documentation

6.63.2.1 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.

6.63.3 Constructor & Destructor Documentation

6.63.3.1 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.

6.63.4 Member Function Documentation

6.63.4.1 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 >::rhs.

6.63.5 Member Data Documentation

6.63.5.1 template<int RTYPE, bool NA> STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs

Definition at line 37 of file clamp.h.

Referenced by Rcpp::clamp().

6.63.5.2 template<int RTYPE, bool NA> STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs

Definition at line 37 of file clamp.h.

Referenced by Rcpp::clamp(), Rcpp::sugar::clamp_operator< RTYPE, NA >::operator()(), and Rcpp::sugar::clamp_↩operator<REALSXP, true >::operator()().

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

6.64 Rcpp::sugar::clamp_operator<REALSXP, true > Struct Template Reference

`#include <clamp.h>`

Public Member Functions

- clamp_operator (double lhs_, double rhs_)
- double operator() (double x) const

Public Attributes

- double lhs
- double rhs

6.64.1 Detailed Description

template<>
struct Rcpp::sugar::clamp_operator<REALSXP, true >

Definition at line 41 of file clamp.h.

6.64.2 Constructor & Destructor Documentation

6.64.2.1 Rcpp::sugar::clamp_operator<REALSXP, true >::clamp_operator (double *lhs_*, double *rhs_*) [inline]

Definition at line 42 of file clamp.h.

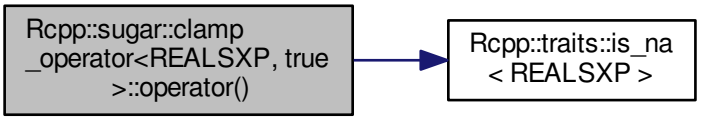
6.64.3 Member Function Documentation

6.64.3.1 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 >(), and Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs.

Here is the call graph for this function:



6.64.4 Member Data Documentation

6.64.4.1 double Rcpp::sugar::clamp_operator<REALSXP, true >::lhs

Definition at line 48 of file clamp.h.

6.64.4.2 double Rcpp::sugar::clamp_operator<REALSXP, true >::rhs

Definition at line 48 of file clamp.h.

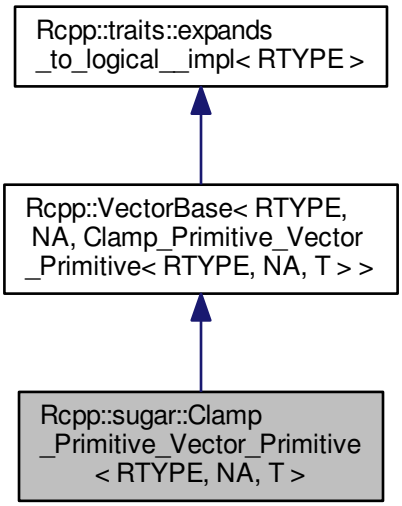
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

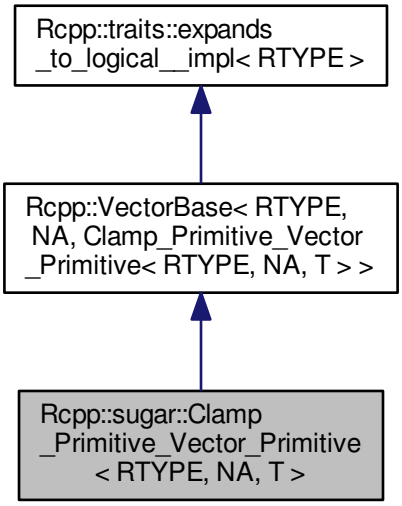
6.65 Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > Class Template Reference

`#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

6.65.1 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.

6.65.2 Member Typedef Documentation

6.65.2.1 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.

6.65.2.2 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.

6.65.3 Constructor & Destructor Documentation

6.65.3.1 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.

6.65.4 Member Function Documentation

6.65.4.1 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.

6.65.4.2 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.

6.65.5 Member Data Documentation

6.65.5.1 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.

6.65.5.2 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.

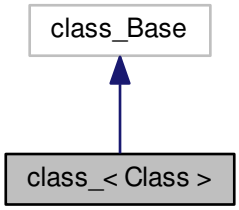
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

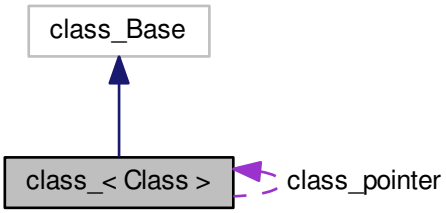
6.66 class_< Class > Class Template Reference

`#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 CppFinalizer< 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 typeinfo_name

Private Member Functions

- self * get_instance ()

6.66.1 Detailed Description

template<typename Class>
class class_< Class >

Definition at line 26 of file class.h.

6.66.2 Member Typedef Documentation

6.66.2.1 template<typename Class > typedef Constructor_Base<Class> class_< Class >::constructor_class

Definition at line 42 of file class.h.

6.66.2.2 template<typename Class > typedef Factory_Base<Class> class_< Class >::factory_class

Definition at line 46 of file class.h.

6.66.2.3 template<typename Class > typedef CppFinalizer<Class> class_< Class >::finalizer_class

Definition at line 40 of file class.h.

6.66.2.4 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.

6.66.2.5 template<typename Class > typedef CppMethod<Class> class_< Class >::method_class

Definition at line 29 of file class.h.

6.66.2.6 template<typename Class > typedef std::map<std::string,method_class*> class_< Class >::METHOD_MAP

Definition at line 36 of file class.h.

6.66.2.7 template<typename Class > typedef std::pair<const std::string,method_class*> class_< Class >::PAIR

Definition at line 37 of file class.h.

6.66.2.8 template<typename Class > typedef CppProperty<Class> class_< Class >::prop_class

Definition at line 50 of file class.h.

6.66.2.9 template<typename Class > typedef std::pair<const std::string,prop_class*> class_< Class >::PROP_PAIR

Definition at line 52 of file class.h.

6.66.2.10 template<typename Class > typedef std::map<std::string,prop_class*> class_< Class >::PROPERTY_MAP

Definition at line 51 of file class.h.

6.66.2.11 template<typename Class > typedef class_<Class> class_< Class >::self

Definition at line 28 of file class.h.

6.66.2.12 template<typename Class > typedef SignedConstructor<Class> class_< Class >::signed_constructor_class

Definition at line 43 of file class.h.

6.66.2.13 template<typename Class > typedef SignedFactory<Class> class_< Class >::signed_factory_class

Definition at line 47 of file class.h.

6.66.2.14 template<typename Class > typedef SignedMethod<Class> class_< Class >::signed_method_class

Definition at line 31 of file class.h.

6.66.2.15 template<typename Class > typedef std::vector<signed_constructor_class*> class_< Class >::vec_signed_constructor

Definition at line 44 of file class.h.

6.66.2.16 template<typename Class > typedef std::vector<signed_factory_class*> class_< Class >::vec_signed_factory

Definition at line 48 of file class.h.

6.66.2.17 template<typename Class > typedef std::vector<signed_method_class*> class_< Class >::vec_signed_method

Definition at line 32 of file class.h.

6.66.2.18 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.

6.66.2.19 template<typename Class > typedef Rcpp::XPtr<Class> class_< Class >::XP

Definition at line 39 of file class.h.

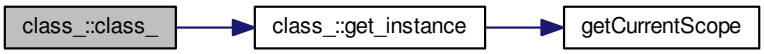
6.66.3 Constructor & Destructor Documentation

6.66.3.1 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:



6.66.3.2 template<typename Class > class_< Class >::~class_ ( ) [inline]

Definition at line 98 of file class.h.

6.66.3.3 template<typename Class > class_< Class >::class_ ( ) [inline]

Definition at line 481 of file class.h.

6.66.4 Member Function Documentation

6.66.4.1 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.

6.66.4.2 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.

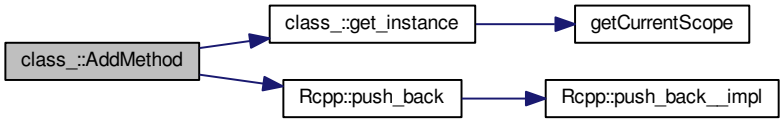
6.66.4.3 template<typename Class > self& class_< Class >::AddMethod (const char * *name_*, method_class * *m*, ValidMethod *valid* = &yes, const char * *docstring* = 0) [inline]

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:



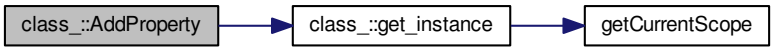
6.66.4.4 template<typename Class > self& class_< Class >::AddProperty (const char * *name_*, prop_class * *p*) [inline]

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:



6.66.4.5 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.

6.66.4.6 template<typename Class > self& class_< Class >::default_constructor (const char * *docstring* = 0, ValidConstructor *valid* = &yes_arity<0>) [inline]

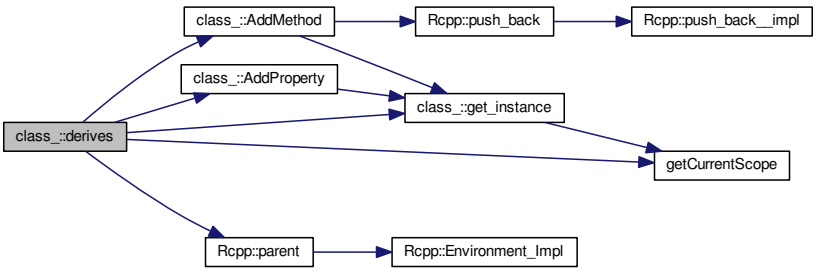
Definition at line 110 of file class.h.

6.66.4.7 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(), getCurrentScope(), Rcpp::parent(), and class_< Class >::properties.

Here is the call graph for this function:

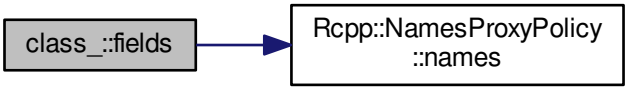


6.66.4.8 template<typename Class > Rcpp::List class_< Class >::fields (const XP_Class & *class_xp*) [inline]

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:



6.66.4.9 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:

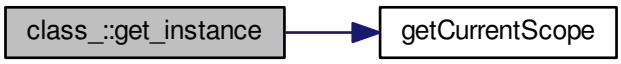
6.66.4.10 template<typename Class > self* class_< Class >::get_instance ( ) [inline], [private]

Definition at line 70 of file class.h.

References class_< Class >::class_pointer, DEMANGLE, class_< Class >::finalizer_pointer, getCurrentScope(), R↩ CPP_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:



6.66.4.11 template<typename Class > std::string class_< Class >::get_typeinfo_name ( ) [inline]

Definition at line 119 of file class.h.

References class_< Class >::typeinfo_name.

6.66.4.12 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.

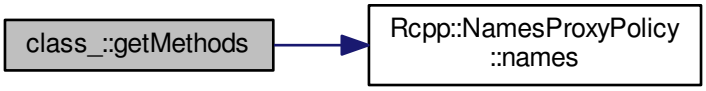
References class_< Class >::constructors.

6.66.4.13 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:



6.66.4.14 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.

6.66.4.15 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.

6.66.4.16 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.

6.66.4.17 template<typename Class > bool class_< Class >::has_property (const std::string & *m*) [inline]

Definition at line 262 of file class.h.

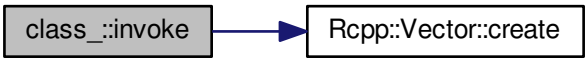
References class_< Class >::properties.

6.66.4.18 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< RTYPE, StoragePolicy >::create(), and END_RCPP.

Here is the call graph for this function:



6.66.4.19 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.

6.66.4.20 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.

6.66.4.21 template<typename Class > Rcpp::CharacterVector class_< Class >::method_names ( ) [inline]

Definition at line 276 of file class.h.

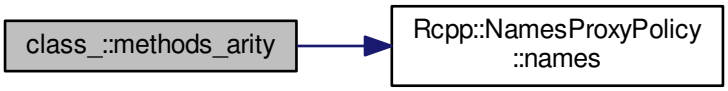
References class_< Class >::vec_methods.

6.66.4.22 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:

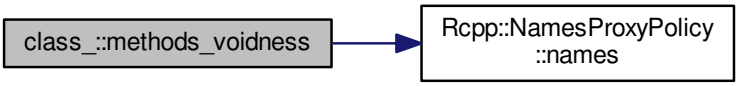


6.66.4.23 template<typename Class > Rcpp::LogicalVector class_< Class >::methods_voidness ( ) [inline]

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:



6.66.4.24 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.

6.66.4.25 template<typename Class > std::string class_< Class >::property_class (const std::string & *p*) [inline]

Definition at line 270 of file class.h.

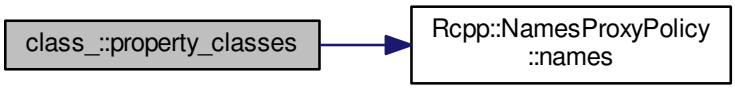
References class_< Class >::properties.

6.66.4.26 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:



6.66.4.27 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.

6.66.4.28 template<typename Class > Rcpp::CharacterVector class_< Class >::property_names ( ) [inline]

Definition at line 345 of file class.h.

References class_< Class >::properties.

6.66.4.29 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.

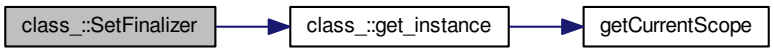
6.66.4.30 template<typename Class > void class_< Class >::SetFinalizer (finalizer_class * f) [inline]

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:



6.66.4.31 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.

6.66.5 Member Data Documentation

6.66.5.1 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().

6.66.5.2 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().

6.66.5.3 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().

6.66.5.4 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().

6.66.5.5 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 >::derives(), 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().

6.66.5.6 template<typename Class > int class_< Class >::specials

Definition at line 474 of file class.h.

Referenced by class_< Class >::complete().

6.66.5.7 template<typename Class > std::string class_< Class >::typeinfo_name

Definition at line 478 of file class.h.

Referenced by class_< Class >::get_instance(), and class_< Class >::get_typeinfo_name().

6.66.5.8 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

6.67 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

6.67.1 Detailed Description

Definition at line 26 of file class_Base.h.

6.67.2 Member Typedef Documentation

6.67.2.1 typedef std::map< std::string, int > Rcpp::class_Base::ENUM

Definition at line 88 of file class_Base.h.

6.67.2.2 typedef std::map< std::string, ENUM > Rcpp::class_Base::ENUM_MAP

Definition at line 89 of file class_Base.h.

6.67.2.3 typedef ENUM_MAP::value_type Rcpp::class_Base::ENUM_MAP_PAIR

Definition at line 90 of file class_Base.h.

6.67.2.4 typedef Rcpp::XPtr<class_Base> Rcpp::class_Base::XP_Class

Definition at line 28 of file class_Base.h.

6.67.3 Constructor & Destructor Documentation

6.67.3.1 Rcpp::class_Base::class_Base ( ) [inline]

Definition at line 30 of file class_Base.h.

6.67.3.2 Rcpp::class_Base::class_Base (const char * *name_*, const char * *doc*) [inline]

Definition at line 31 of file class_Base.h.

6.67.3.3 virtual Rcpp::class_Base::~class_Base ( ) [inline], [virtual]

Definition at line 69 of file class_Base.h.

6.67.4 Member Function Documentation

6.67.4.1 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().

6.67.4.2 virtual Rcpp::CharacterVector Rcpp::class_Base::complete ( ) [inline], [virtual]

Definition at line 68 of file class_Base.h.

6.67.4.3 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().

6.67.4.4 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().

6.67.4.5 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().

6.67.4.6 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().

6.67.4.7 virtual SEXP Rcpp::class_Base::getProperty (SEXP , SEXP) [inline], [virtual]

Definition at line 71 of file class_Base.h.

6.67.4.8 virtual bool Rcpp::class_Base::has_default_constructor ( ) [inline], [virtual]

Definition at line 40 of file class_Base.h.

6.67.4.9 virtual bool Rcpp::class_Base::has_method (const std::string &) [inline], [virtual]

Definition at line 41 of file class_Base.h.

6.67.4.10 virtual bool Rcpp::class_Base::has_property (const std::string &) [inline], [virtual]

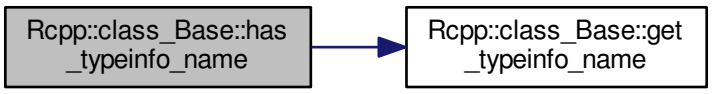
Definition at line 44 of file class_Base.h.

6.67.4.11 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:



6.67.4.12 virtual SEXP Rcpp::class_Base::invoke (SEXP , SEXP , SEXP * , int) [inline], [virtual]

Definition at line 50 of file class_Base.h.

6.67.4.13 virtual SEXP Rcpp::class_Base::invoke_notvoid (SEXP , SEXP , SEXP * , int) [inline], [virtual]

Definition at line 56 of file class_Base.h.

6.67.4.14 virtual SEXP Rcpp::class_Base::invoke_void (SEXP , SEXP , SEXP * , int) [inline], [virtual]

Definition at line 53 of file class_Base.h.

6.67.4.15 virtual Rcpp::CharacterVector Rcpp::class_Base::method_names ( ) [inline], [virtual]

Definition at line 60 of file class_Base.h.

6.67.4.16 virtual Rcpp::IntegerVector Rcpp::class_Base::methods_arity ( ) [inline], [virtual]

Definition at line 64 of file class_Base.h.

6.67.4.17 virtual Rcpp::LogicalVector Rcpp::class_Base::methods_voidness ( ) [inline], [virtual]

Definition at line 65 of file class_Base.h.

6.67.4.18 virtual SEXP Rcpp::class_Base::newInstance (SEXP *, int) [inline], [virtual]

Definition at line 47 of file class_Base.h.

6.67.4.19 virtual std::string Rcpp::class_Base::property_class (const std::string &) [inline], [virtual]

Definition at line 63 of file class_Base.h.

6.67.4.20 virtual Rcpp::List Rcpp::class_Base::property_classes ( ) [inline], [virtual]

Definition at line 66 of file class_Base.h.

6.67.4.21 virtual bool Rcpp::class_Base::property_is_readonly (const std::string &) [inline], [virtual]

Definition at line 62 of file class_Base.h.

6.67.4.22 virtual Rcpp::CharacterVector Rcpp::class_Base::property_names ( ) [inline], [virtual]

Definition at line 61 of file class_Base.h.

6.67.4.23 virtual void Rcpp::class_Base::run_finalizer (SEXP) [inline], [virtual]

Definition at line 38 of file class_Base.h.

6.67.4.24 virtual void Rcpp::class_Base::setProperty (SEXP , SEXP , SEXP) [inline], [virtual]

Definition at line 74 of file class_Base.h.

6.67.5 Member Data Documentation

6.67.5.1 std::string Rcpp::class_Base::docstring

Definition at line 86 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.67.5.2 ENUM_MAP Rcpp::class_Base::enums

Definition at line 91 of file class_Base.h.

Referenced by add_enum(), and Rcpp::CppClass::CppClass().

6.67.5.3 std::string Rcpp::class_Base::name

Definition at line 85 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.67.5.4 std::vector<std::string> Rcpp::class_Base::parents

Definition at line 92 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

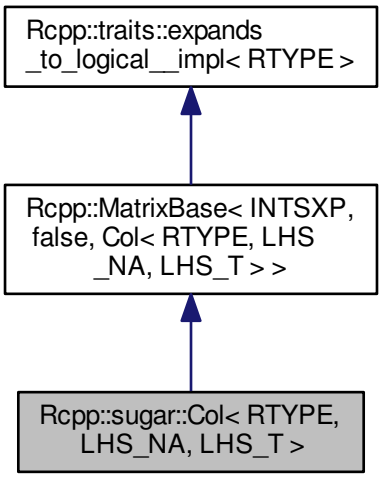
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/class_Base.h

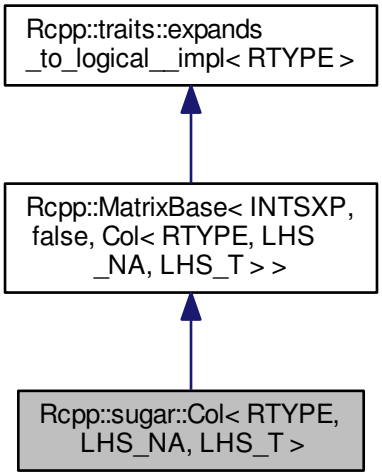
6.68 Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T > Class Template Reference

`#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)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Attributes

- int nr
- int nc

6.68.1 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.

6.68.2 Member Typedef Documentation

6.68.2.1 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.

6.68.3 Constructor & Destructor Documentation

6.68.3.1 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.

6.68.4 Member Function Documentation

6.68.4.1 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.

6.68.4.2 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.

6.68.4.3 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.

6.68.4.4 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.

6.68.5 Member Data Documentation

6.68.5.1 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, L↩ HS_T >::size().

6.68.5.2 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, L↩ HS_T >::size().

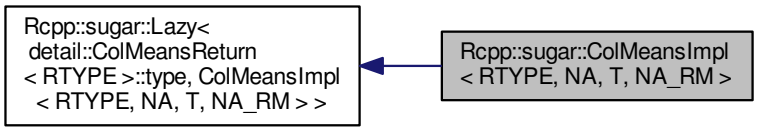
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/col.h

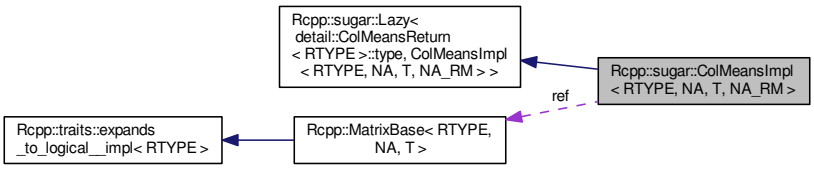
6.69 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference

`#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

6.69.1 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.

6.69.2 Member Typedef Documentation

6.69.2.1 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.

6.69.2.2 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.

6.69.2.3 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.

6.69.3 Constructor & Destructor Documentation

6.69.3.1 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.

6.69.4 Member Function Documentation

6.69.4.1 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.

6.69.5 Member Data Documentation

6.69.5.1 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.

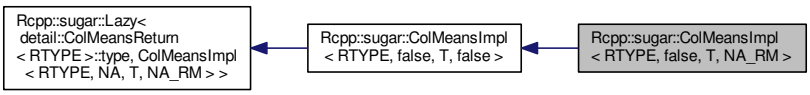
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

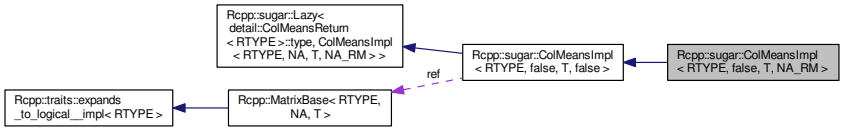
6.70 Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference

`#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

6.70.1 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.

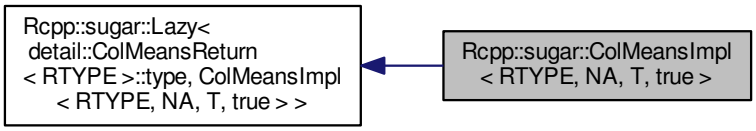
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

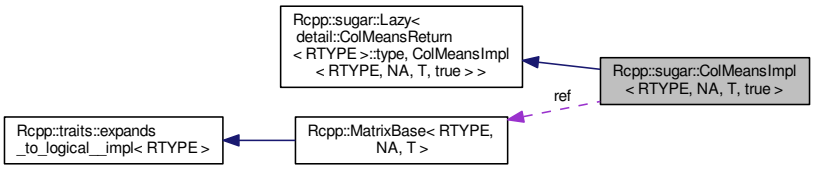
6.71 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true > Class Template Reference

`#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

6.71.1 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.

6.71.2 Member Typedef Documentation

6.71.2.1 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.

6.71.2.2 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.

6.71.2.3 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.

6.71.3 Constructor & Destructor Documentation

6.71.3.1 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.

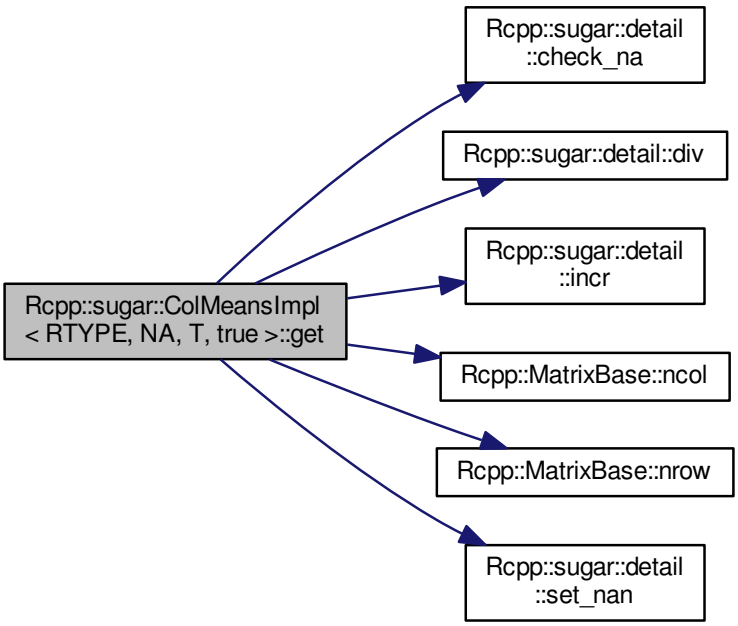
6.71.4 Member Function Documentation

6.71.4.1 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(), and Rcpp::sugar::detail::set_nan().

Here is the call graph for this function:



6.71.5 Member Data Documentation

6.71.5.1 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.

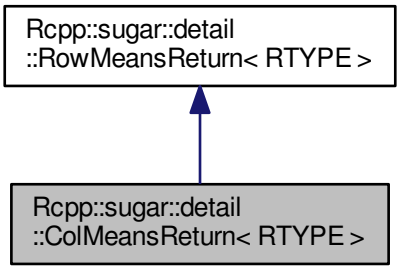
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

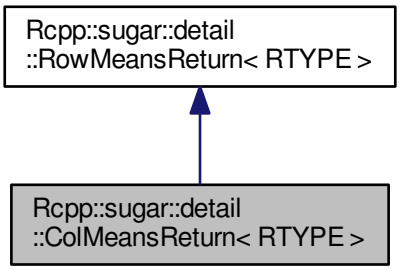
6.72 Rcpp::sugar::detail::ColMeansReturn< RTYPE > Struct Template Reference

`#include <rowSums.h>`

Inheritance diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



Additional Inherited Members

6.72.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::ColMeansReturn< RTYPE >

Definition at line 115 of file rowSums.h.

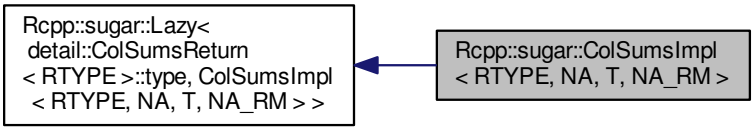
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

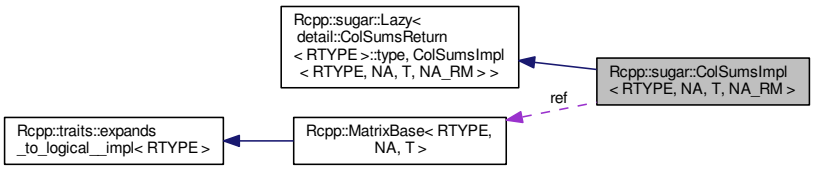
6.73 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference

`#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

6.73.1 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.

6.73.2 Member Typedef Documentation

6.73.2.1 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.

6.73.2.2 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.

6.73.2.3 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.

6.73.3 Constructor & Destructor Documentation

6.73.3.1 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.

6.73.4 Member Function Documentation

6.73.4.1 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.

6.73.5 Member Data Documentation

6.73.5.1 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.

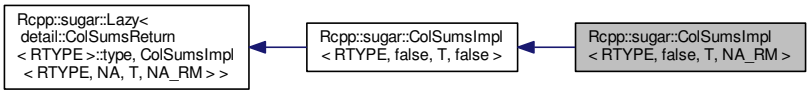
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

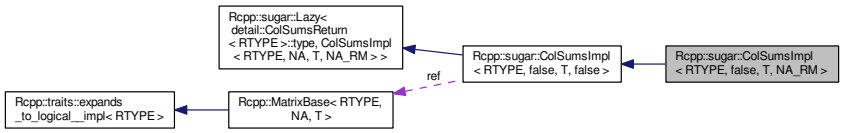
6.74 Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference

`#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

6.74.1 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.

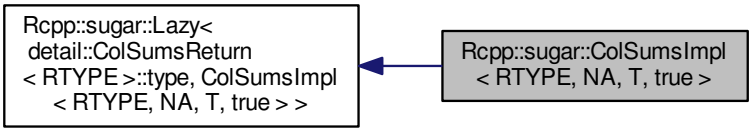
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

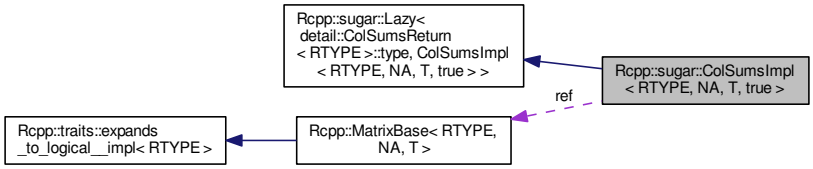
6.75 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true > Class Template Reference

`#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 >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.75.1 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.

6.75.2 Member Typedef Documentation

6.75.2.1 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.

6.75.2.2 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.

6.75.2.3 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.

6.75.3 Constructor & Destructor Documentation

6.75.3.1 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.

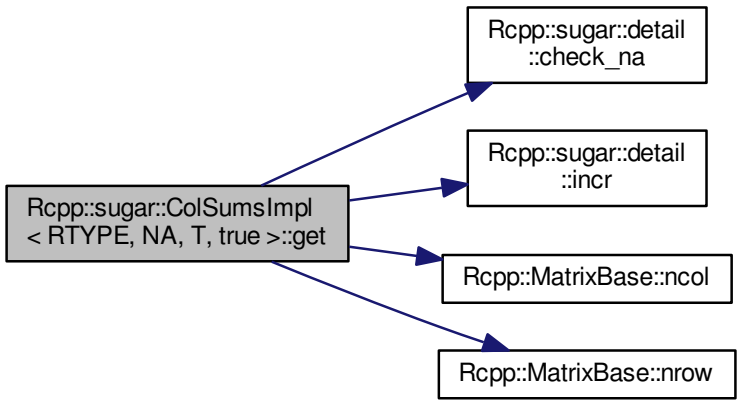
6.75.4 Member Function Documentation

6.75.4.1 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(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow().

Here is the call graph for this function:



6.75.5 Member Data Documentation

6.75.5.1 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.

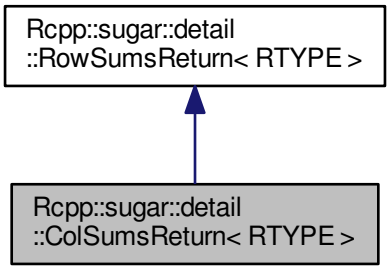
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

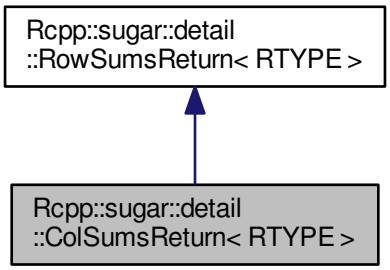
6.76 Rcpp::sugar::detail::ColSumsReturn< RTYPE > Struct Template Reference

`#include <rowSums.h>`

Inheritance diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



Additional Inherited Members

6.76.1 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

6.77 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_

6.77.1 Detailed Description

Definition at line 429 of file attributes.cpp.

6.77.2 Constructor & Destructor Documentation

6.77.2.1 Rcpp::attributes::CommentState::CommentState ( ) [inline]

Definition at line 431 of file attributes.cpp.

6.77.2.2 Rcpp::attributes::CommentState::CommentState (const CommentState &) [private]

6.77.3 Member Function Documentation

6.77.3.1 bool Rcpp::attributes::CommentState::inComment ( ) const [inline]

Definition at line 437 of file attributes.cpp.

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

6.77.3.2 CommentState& Rcpp::attributes::CommentState::operator= (const CommentState &) [private]

6.77.3.3 void Rcpp::attributes::CommentState::reset ( ) [inline]

Definition at line 439 of file attributes.cpp.

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

6.77.3.4 void Rcpp::attributes::CommentState::submitLine (const std::string & *line*)

Definition at line 1683 of file attributes.cpp.

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

6.77.4 Member Data Documentation

6.77.4.1 bool Rcpp::attributes::CommentState::inComment_ [private]

Definition at line 441 of file attributes.cpp.

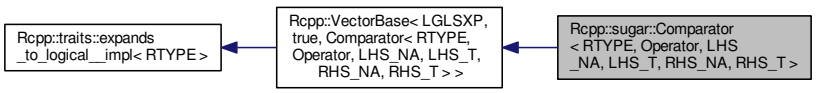
The documentation for this class was generated from the following file:

- src/attributes.cpp

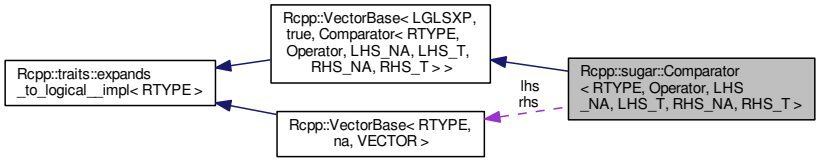
6.78 Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.78.1 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.

6.78.2 Member Typedef Documentation

6.78.2.1 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.

6.78.2.2 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.

6.78.2.3 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.

6.78.3 Constructor & Destructor Documentation

6.78.3.1 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.

6.78.4 Member Function Documentation

6.78.4.1 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.

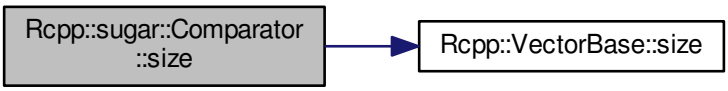
References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.78.4.2 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:



6.78.5 Member Data Documentation

6.78.5.1 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().

6.78.5.2 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[]().

6.78.5.3 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[]().

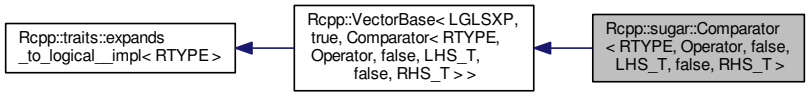
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

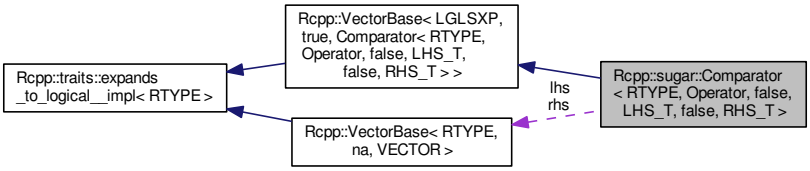
6.79 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.79.1 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.

6.79.2 Member Typedef Documentation

6.79.2.1 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.

6.79.2.2 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.

6.79.2.3 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.

6.79.3 Constructor & Destructor Documentation

6.79.3.1 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.

6.79.4 Member Function Documentation

6.79.4.1 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.

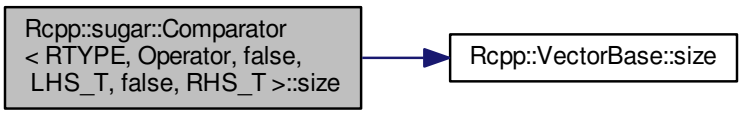
References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.79.4.2 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:



6.79.5 Member Data Documentation

6.79.5.1 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.

6.79.5.2 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.

6.79.5.3 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.

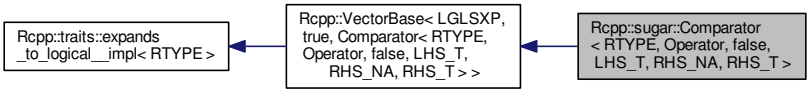
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

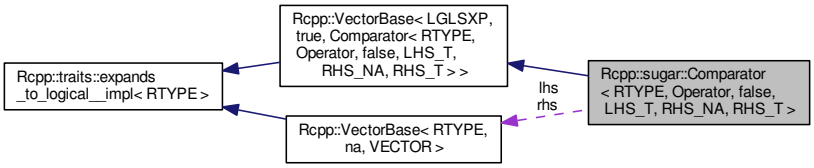
6.80 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.80.1 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.

6.80.2 Member Typedef Documentation

6.80.2.1 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.

6.80.2.2 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.

6.80.2.3 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.

6.80.3 Constructor & Destructor Documentation

6.80.3.1 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.

6.80.4 Member Function Documentation

6.80.4.1 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.

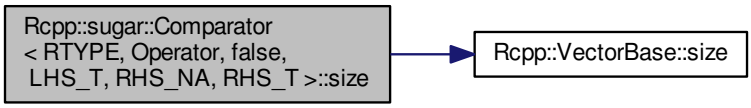
References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.80.4.2 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:



6.80.5 Member Data Documentation

6.80.5.1 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.

6.80.5.2 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.

6.80.5.3 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.

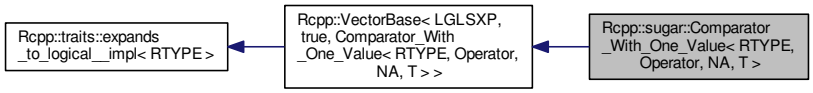
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

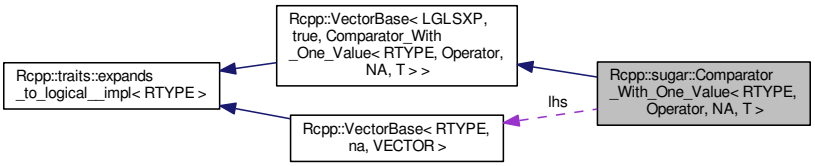
6.81 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T > Class Template Reference

`#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

6.81.1 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.

6.81.2 Member Typedef Documentation

6.81.2.1 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.

6.81.2.2 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.

6.81.2.3 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.

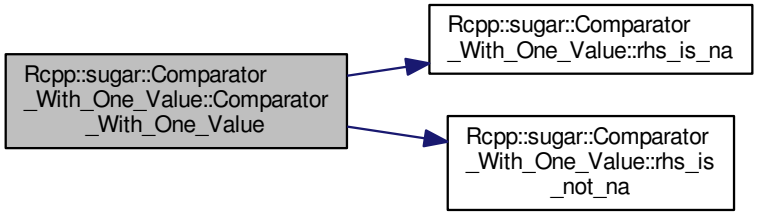
6.81.3 Constructor & Destructor Documentation

6.81.3.1 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:



6.81.4 Member Function Documentation

6.81.4.1 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.

6.81.4.2 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().

6.81.4.3 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 >::op.

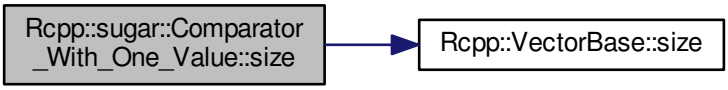
Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value().

6.81.4.4 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:



6.81.5 Member Data Documentation

6.81.5.1 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 >::size(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size().

6.81.5.2 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[]().

6.81.5.3 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().

6.81.5.4 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(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_na().

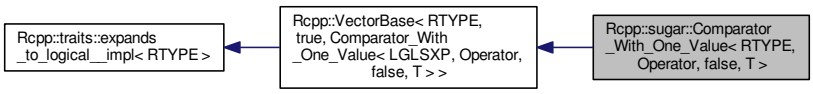
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h

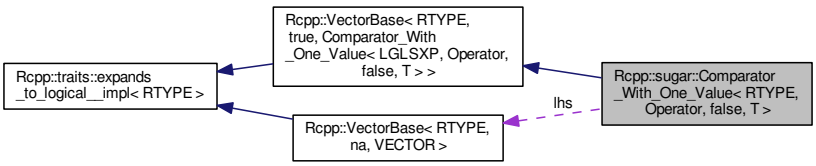
6.82 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T > Class Template Reference

`#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

6.82.1 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.

6.82.2 Member Typedef Documentation

6.82.2.1 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.

6.82.2.2 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.

6.82.2.3 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.

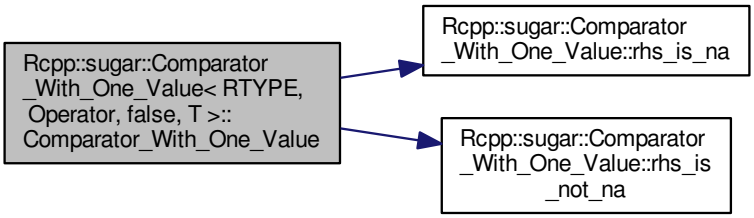
6.82.3 Constructor & Destructor Documentation

6.82.3.1 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:



6.82.4 Member Function Documentation

6.82.4.1 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.

6.82.4.2 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.

6.82.4.3 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.

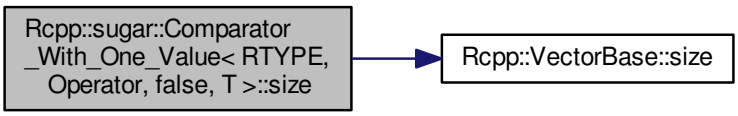
References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::op.

6.82.4.4 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:



6.82.5 Member Data Documentation

6.82.5.1 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.

6.82.5.2 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.

6.82.5.3 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.

6.82.5.4 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.

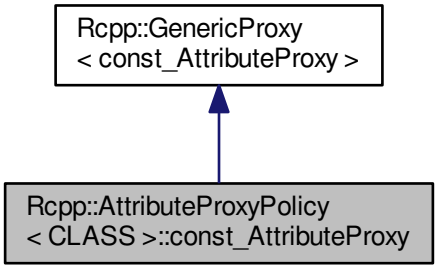
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h

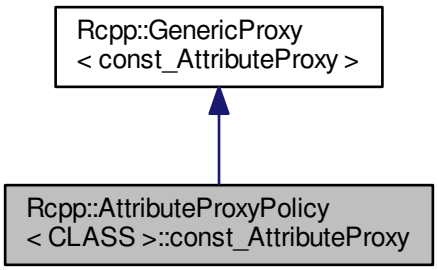
6.83 Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy Class Reference

`#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

6.83.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy

Definition at line 56 of file AttributeProxy.h.

6.83.2 Constructor & Destructor Documentation

6.83.2.1 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.

6.83.3 Member Function Documentation

6.83.3.1 template<typename CLASS> SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::get ( ) const [inline], [private]

Definition at line 68 of file AttributeProxy.h.

6.83.3.2 template<typename CLASS > Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator SEXP ( ) const [inline]

Definition at line 55 of file proxy.h.

6.83.3.3 template<typename CLASS > template<typename T > Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator T ( ) const

Definition at line 50 of file proxy.h.

6.83.4 Member Data Documentation

6.83.4.1 template<typename CLASS> SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::attr_name [private]

Definition at line 66 of file AttributeProxy.h.

6.83.4.2 template<typename CLASS> const CLASS& Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::parent [private]

Definition at line 65 of file AttributeProxy.h.

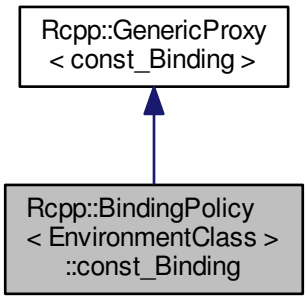
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/AttributeProxy.h
- inst/include/Rcpp/api/meat/proxy.h

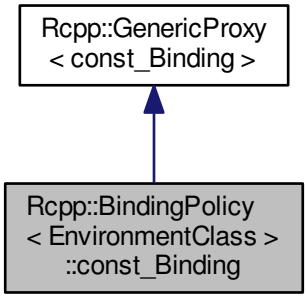
6.84 Rcpp::BindingPolicy< EnvironmentClass >::const_Binding Class Reference

`#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

6.84.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::const_Binding

Definition at line 72 of file Binding.h.

6.84.2 Constructor & Destructor Documentation

6.84.2.1 template<typename EnvironmentClass > Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::const_Binding (const EnvironmentClass & *env_*, const std::string & *name_*) [inline]

Definition at line 74 of file Binding.h.

6.84.3 Member Function Documentation

6.84.3.1 template<typename EnvironmentClass > bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::active ( ) const [inline]

Definition at line 77 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.84.3.2 template<typename EnvironmentClass > bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists ( ) const [inline]

Definition at line 83 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.84.3.3 template<typename EnvironmentClass > SEXP Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::get ( ) const [inline], [private]

Definition at line 90 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.84.3.4 template<typename EnvironmentClass > bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::locked ( ) const [inline]

Definition at line 80 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.84.3.5 template<typename CLASS > template<typename T > Rcpp::BindingPolicy< CLASS >::const_Binding::operator T ( ) const

Definition at line 143 of file proxy.h.

6.84.4 Member Data Documentation

6.84.4.1 template<typename EnvironmentClass > const EnvironmentClass& Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env [private]

Definition at line 94 of file Binding.h.

6.84.4.2 template<typename EnvironmentClass > std::string Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name [private]

Definition at line 95 of file Binding.h.

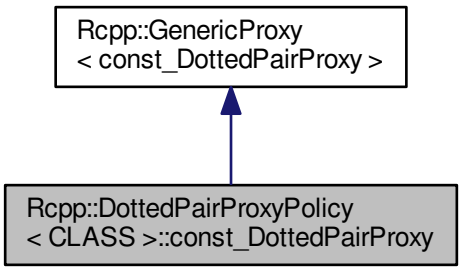
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/Binding.h
- inst/include/Rcpp/api/meat/proxy.h

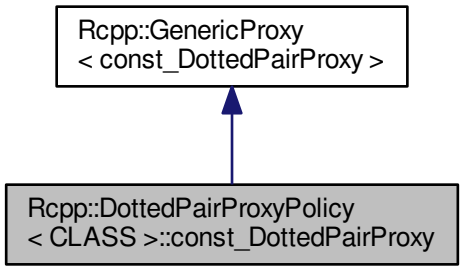
6.85 Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy Class Reference

`#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

6.85.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy

Definition at line 72 of file DottedPairProxy.h.

6.85.2 Constructor & Destructor Documentation

6.85.2.1 template<typename CLASS > Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::const_DottedPairProxy (const CLASS & *v*, int *index_*) [inline]

Definition at line 74 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.85.3 Member Function Documentation

6.85.3.1 template<typename CLASS > SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::get ( ) const [inline]

Definition at line 83 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.85.3.2 template<typename CLASS > Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator SEXP ( ) const [inline]

Definition at line 86 of file DottedPairProxy.h.

6.85.3.3 template<typename CLASS > template<typename T > Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator T ( ) const

Definition at line 171 of file proxy.h.

6.85.4 Member Data Documentation

6.85.4.1 template<typename CLASS > SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::node [private]

Definition at line 91 of file DottedPairProxy.h.

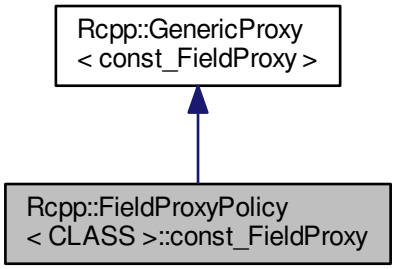
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/DottedPairProxy.h
- inst/include/Rcpp/api/meat/proxy.h

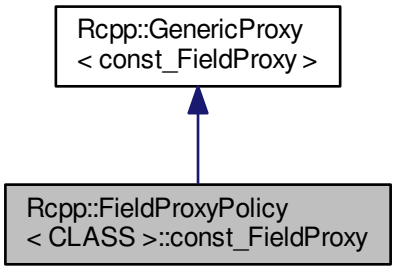
6.86 Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy Class Reference

`#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

6.86.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy

Definition at line 54 of file FieldProxy.h.

6.86.2 Constructor & Destructor Documentation

6.86.2.1 template<typename CLASS > Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::const_FieldProxy (const CLASS & *v*, const std::string & *name*) [inline]

Definition at line 56 of file FieldProxy.h.

6.86.3 Member Function Documentation

6.86.3.1 template<typename CLASS > SEXP Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get ( ) const [inline], [private]

Definition at line 68 of file FieldProxy.h.

References Rcpp::Rcpp_eval().

Here is the call graph for this function:

6.86.3.2 template<typename CLASS > Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator SEXP ( ) const [inline]

Definition at line 60 of file FieldProxy.h.

6.86.3.3 template<typename CLASS > template<typename T > Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator T ( ) const

Definition at line 201 of file proxy.h.

6.86.4 Member Data Documentation

6.86.4.1 template<typename CLASS > const std::string& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::field_name [private]

Definition at line 66 of file FieldProxy.h.

6.86.4.2 template<typename CLASS > const CLASS& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::parent [private]

Definition at line 65 of file FieldProxy.h.

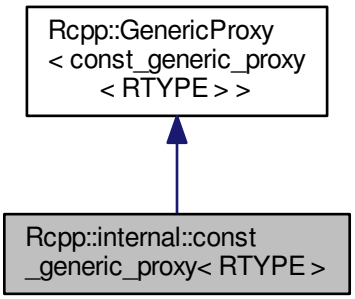
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/FieldProxy.h
- inst/include/Rcpp/api/meat/proxy.h

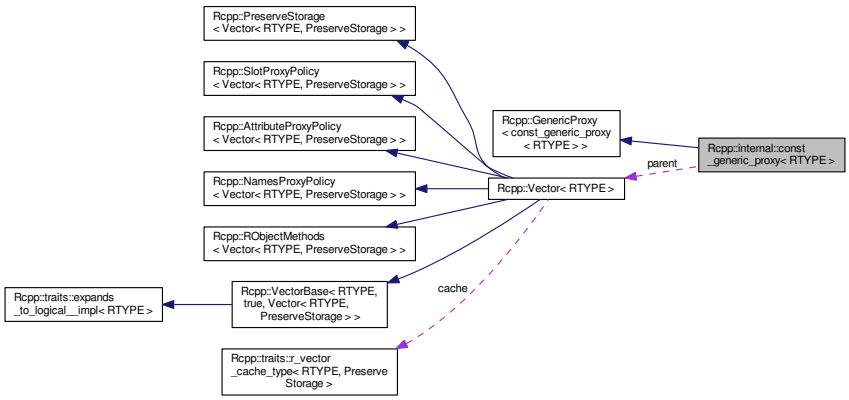
6.87 Rcpp::internal::const_generic_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Inheritance diagram for Rcpp::internal::const_generic_proxy< RTYPE >:



Collaboration diagram for Rcpp::internal::const_generic_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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

6.87.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::const_generic_proxy< RTYPE >

Definition at line 31 of file 00_forward_proxy.h.

6.87.2 Member Typedef Documentation

6.87.2.1 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::VECTOR

Definition at line 29 of file const_generic_proxy.h.

6.87.3 Constructor & Destructor Documentation

6.87.3.1 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::const_generic_proxy ( ) [inline]

Definition at line 31 of file const_generic_proxy.h.

6.87.3.2 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::const_generic_proxy (const const_generic_proxy< RTYPE > & *other*) [inline]

Definition at line 33 of file const_generic_proxy.h.

6.87.3.3 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::const_generic_proxy (const VECTOR & *v*, R_xlen_t *i*) [inline]

Definition at line 36 of file const_generic_proxy.h.

6.87.4 Member Function Documentation

6.87.4.1 template<int RTYPE> SEXP Rcpp::internal::const_generic_proxy< RTYPE >::get ( ) const [inline], [private]

Definition at line 57 of file const_generic_proxy.h.

6.87.4.2 template<int RTYPE> void Rcpp::internal::const_generic_proxy< RTYPE >::move (R_xlen_t *n*) [inline]

Definition at line 50 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE >::index.

6.87.4.3 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::operator bool ( ) const [inline]

Definition at line 47 of file const_generic_proxy.h.

6.87.4.4 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::operator int ( ) const [inline]

Definition at line 48 of file const_generic_proxy.h.

6.87.4.5 template<int RTYPE> Rcpp::internal::const_generic_proxy< RTYPE >::operator SEXP ( ) const [inline]

Definition at line 38 of file const_generic_proxy.h.

6.87.4.6 template<int RTYPE> template<typename U > Rcpp::internal::const_generic_proxy< RTYPE >::operator U ( ) const [inline]

Definition at line 42 of file const_generic_proxy.h.

6.87.5 Member Data Documentation

6.87.5.1 template<int RTYPE> R_xlen_t Rcpp::internal::const_generic_proxy< RTYPE >::index

Definition at line 53 of file const_generic_proxy.h.

Referenced by Rcpp::internal::const_generic_proxy< RTYPE >::move().

6.87.5.2 template<int RTYPE> const VECTOR* Rcpp::internal::const_generic_proxy< RTYPE >::parent

Definition at line 52 of file const_generic_proxy.h.

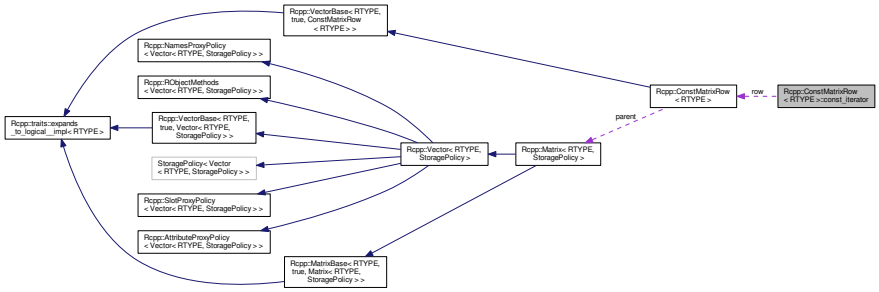
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/const_generic_proxy.h

6.88 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

6.88.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >::const_iterator

Definition at line 181 of file MatrixRow.h.

6.88.2 Member Typedef Documentation

6.88.2.1 template<int RTYPE> typedef int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::difference_type

Definition at line 185 of file MatrixRow.h.

6.88.2.2 template<int RTYPE> typedef std::random_access_iterator_tag Rcpp::ConstMatrixRow< RTYPE >::const_iterator::iterator_category

Definition at line 190 of file MatrixRow.h.

6.88.2.3 template<int RTYPE> typedef std::iterator_traits<vector_iterator>::pointer Rcpp::ConstMatrixRow< RTYPE >::const_iterator::pointer

Definition at line 188 of file MatrixRow.h.

6.88.2.4 template<int RTYPE> typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::reference

Definition at line 187 of file MatrixRow.h.

6.88.2.5 template<int RTYPE> typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::value_type

Definition at line 186 of file MatrixRow.h.

6.88.2.6 template<int RTYPE> typedef traits::r_vector_iterator<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::vector_iterator

Definition at line 183 of file MatrixRow.h.

6.88.3 Constructor & Destructor Documentation

6.88.3.1 template<int RTYPE> Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (const const_iterator & *other*) [inline]

Definition at line 192 of file MatrixRow.h.

6.88.3.2 template<int RTYPE> Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (const ConstMatrixRow & *row_*, int *index_*) [inline]

Definition at line 193 of file MatrixRow.h.

6.88.4 Member Function Documentation

6.88.4.1 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator!= (const const_iterator & *other*) [inline]

Definition at line 230 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.2 template<int RTYPE> const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator* ( ) [inline]

Definition at line 222 of file MatrixRow.h.

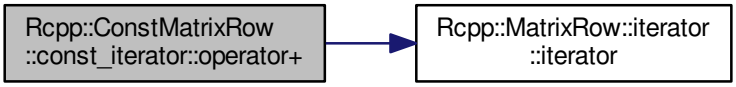
References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.88.4.3 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+ (difference_type *n*) const [inline]

Definition at line 215 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp::MatrixRow< RTYPE >::iterator::row.

Here is the call graph for this function:



6.88.4.4 template<int RTYPE> const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ ( ) [inline]

Definition at line 195 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.5 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ (int) [inline]

Definition at line 199 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.6 template<int RTYPE> const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+= (difference_type *n*) [inline]

Definition at line 219 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.7 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (difference_type *n*) const [inline]

Definition at line 216 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp::MatrixRow< RTYPE >::iterator::row.

Here is the call graph for this function:

6.88.4.8 template<int RTYPE> difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (const const_iterator & *other*) const [inline]

Definition at line 217 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.9 template<int RTYPE> difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (const const_iterator & *other*) [inline]

Definition at line 240 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.10 template<int RTYPE> const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- ( ) [inline]

Definition at line 205 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.11 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- (int) [inline]

Definition at line 209 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.12 template<int RTYPE> const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-= (difference_type *n*) [inline]

Definition at line 220 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.88.4.13 template<int RTYPE> const pointer Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-> ( ) [inline]

Definition at line 225 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.88.4.14 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator< (const const_iterator & *other*) [inline]

Definition at line 231 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.15 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<= (const const_iterator & *other*) [inline]

Definition at line 233 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.16 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator== (const const_iterator & *other*) [inline]

Definition at line 229 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.17 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator> (const const_iterator & *other*) [inline]

Definition at line 232 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.18 template<int RTYPE> bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>= (const const_iterator & *other*) [inline]

Definition at line 234 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.88.4.19 template<int RTYPE> const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[ ] (int *i*) const [inline]

Definition at line 236 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.88.5 Member Data Documentation

6.88.5.1 template<int RTYPE> int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index [private]

Definition at line 246 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>(), and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>=().

6.88.5.2 template<int RTYPE> const ConstMatrixRow& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::row [private]

Definition at line 245 of file MatrixRow.h.

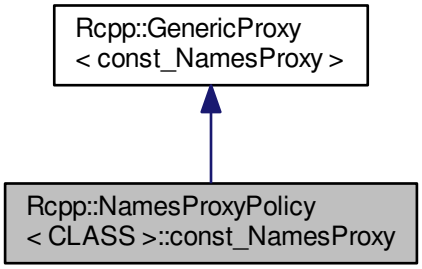
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

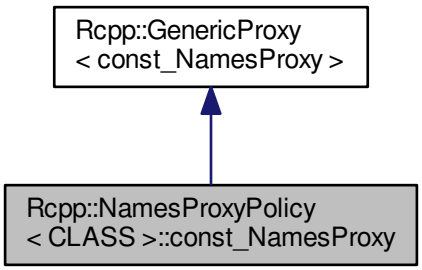
6.89 Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy Class Reference

`#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

6.89.1 Detailed Description

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy

Definition at line 65 of file NamesProxy.h.

6.89.2 Constructor & Destructor Documentation

6.89.2.1 template<typename CLASS> Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::const_NamesProxy (const CLASS & *v*) [inline]

Definition at line 67 of file NamesProxy.h.

6.89.3 Member Function Documentation

6.89.3.1 template<typename CLASS> SEXP Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::get ( ) const [inline], [private]

Definition at line 74 of file NamesProxy.h.

References RCPP_GET_NAMES.

6.89.3.2 template<typename CLASS > template<typename T > Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::operator T ( ) const

Definition at line 76 of file proxy.h.

6.89.4 Member Data Documentation

6.89.4.1 template<typename CLASS> const CLASS& Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::parent [private]

Definition at line 72 of file NamesProxy.h.

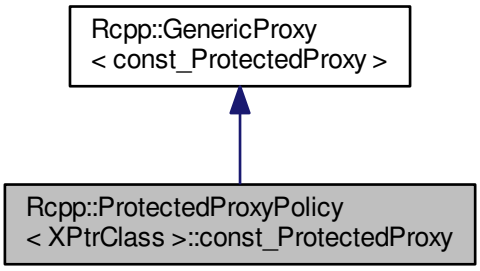
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/NamesProxy.h
- inst/include/Rcpp/api/meat/proxy.h

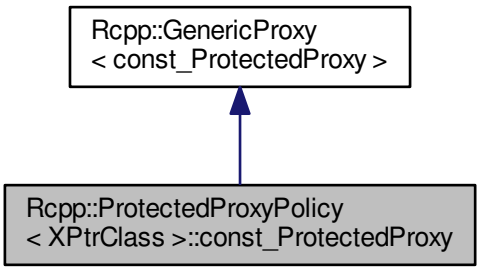
6.90 Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy Class Reference

`#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

6.90.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy

Definition at line 61 of file ProtectedProxy.h.

6.90.2 Constructor & Destructor Documentation

6.90.2.1 template<class XPtrClass> Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::const_Protected↩ Proxy (const XPtrClass & *xp_*) [inline]

Definition at line 63 of file ProtectedProxy.h.

6.90.3 Member Function Documentation

6.90.3.1 template<class XPtrClass> SEXP Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get ( ) const [inline], [private]

Definition at line 77 of file ProtectedProxy.h.

6.90.3.2 template<class XPtrClass> Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator SEXP ( ) const [inline]

Definition at line 70 of file ProtectedProxy.h.

6.90.3.3 template<class XPtrClass> template<typename U > Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator U ( ) const [inline]

Definition at line 66 of file ProtectedProxy.h.

6.90.4 Member Data Documentation

6.90.4.1 template<class XPtrClass> const XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::xp [private]

Definition at line 75 of file ProtectedProxy.h.

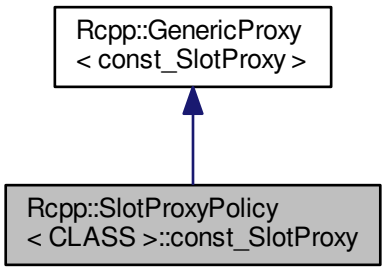
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

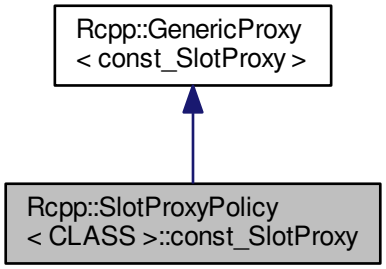
6.91 Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy Class Reference

`#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

6.91.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy

Definition at line 59 of file SlotProxy.h.

6.91.2 Constructor & Destructor Documentation

6.91.2.1 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 >::SlotProxy::slot_name.

6.91.3 Member Function Documentation

6.91.3.1 template<typename CLASS> SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get ( ) const [inline], [private]

Definition at line 78 of file SlotProxy.h.

6.91.3.2 template<typename CLASS> Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator SEXP ( ) const [inline]

Definition at line 70 of file SlotProxy.h.

6.91.3.3 template<typename CLASS> template<typename T > Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator T ( ) const [inline]

Definition at line 67 of file SlotProxy.h.

6.91.4 Member Data Documentation

6.91.4.1 template<typename CLASS> const CLASS& Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::parent [private]

Definition at line 75 of file SlotProxy.h.

6.91.4.2 template<typename CLASS> SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::slot_name [private]

Definition at line 76 of file SlotProxy.h.

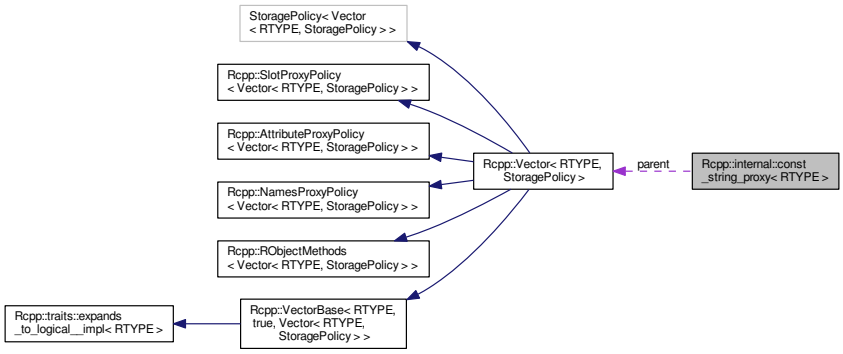
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/SlotProxy.h

6.92 Rcpp::internal::const_string_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Collaboration diagram for Rcpp::internal::const_string_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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>
  std::ostream & operator<< (std::ostream &os, const const_string_proxy< RT > &proxy)
- template<int RT>
  std::string operator+ (const std::string &x, const const_string_proxy< RT > &proxy)

6.92.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::const_string_proxy< RTYPE >

Definition at line 29 of file 00_forward_proxy.h.

6.92.2 Member Typedef Documentation

6.92.2.1 template<int RTYPE> typedef const char* Rcpp::internal::const_string_proxy< RTYPE >::iterator

Definition at line 32 of file const_string_proxy.h.

6.92.2.2 template<int RTYPE> typedef const char& Rcpp::internal::const_string_proxy< RTYPE >::reference

Definition at line 33 of file const_string_proxy.h.

6.92.2.3 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::VECTOR

Definition at line 31 of file const_string_proxy.h.

6.92.3 Constructor & Destructor Documentation

6.92.3.1 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::const_string_proxy ( ) [inline]

Definition at line 35 of file const_string_proxy.h.

6.92.3.2 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::const_string_proxy (const VECTOR & *v*, R_xlen_t *index_*) [inline]

Creates a proxy

Parameters

v	reference to the associated character vector
index	index

Definition at line 43 of file const_string_proxy.h.

6.92.3.3 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::const_string_proxy (SEXP *x*) [inline]

Definition at line 45 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::parent.

6.92.3.4 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::const_string_proxy (const const_string_proxy< RTYPE > & *other*) [inline]

Definition at line 50 of file const_string_proxy.h.

6.92.4 Member Function Documentation

6.92.4.1 template<int RTYPE> iterator Rcpp::internal::const_string_proxy< RTYPE >::begin ( ) const [inline]

Definition at line 93 of file const_string_proxy.h.

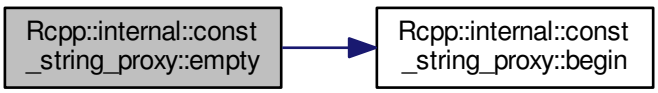
Referenced by Rcpp::internal::const_string_proxy< RTYPE >::empty(), Rcpp::internal::const_string_proxy< RTYP↩E >::end(), Rcpp::internal::const_string_proxy< RTYPE >::operator!=(), Rcpp::internal::const_string_proxy< RTYPE >::operator==(), Rcpp::internal::const_string_proxy< RTYPE >::operator[](), and Rcpp::internal::const_string_proxy< RTYPE >::size().

6.92.4.2 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::empty ( ) const [inline]

Definition at line 96 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:

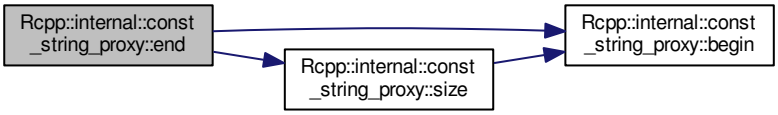


6.92.4.3 template<int RTYPE> iterator Rcpp::internal::const_string_proxy< RTYPE >::end ( ) const [inline]

Definition at line 94 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin(), and Rcpp::internal::const_string_proxy< RTYPE >::size().

Here is the call graph for this function:



6.92.4.4 template<int RTYPE> SEXP Rcpp::internal::const_string_proxy< RTYPE >::get ( ) const [inline]

Definition at line 89 of file const_string_proxy.h.

Referenced by Rcpp::String::operator!=(), and Rcpp::String::operator==().

6.92.4.5 template<int RTYPE> void Rcpp::internal::const_string_proxy< RTYPE >::import (const const_string_proxy< RTYPE > & *other*) [inline]

Definition at line 53 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::index, and Rcpp::internal::const_string_proxy< RTYP↩E >::parent.

6.92.4.6 template<int RTYPE> void Rcpp::internal::const_string_proxy< RTYPE >::move (R_xlen_t *n*) [inline]

Definition at line 87 of file const_string_proxy.h.

6.92.4.7 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::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 71 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::operator+.

6.92.4.8 template<int RTYPE> Rcpp::internal::const_string_proxy< RTYPE >::operator SEXP ( ) const [inline]

rhs use. Retrieves the current value of the element this proxy refers to.

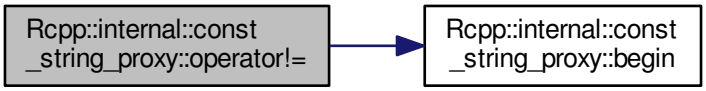
Definition at line 62 of file const_string_proxy.h.

6.92.4.9 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator!= (const char * *other*) [inline]

Definition at line 102 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:

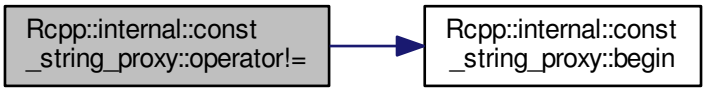


6.92.4.10 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator!= (const const_string_proxy< RTYPE > & *other*) [inline]

Definition at line 109 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:



6.92.4.11 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator!= (SEXP *other*) const [inline]

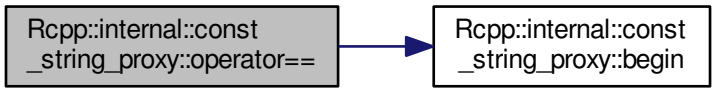
Definition at line 117 of file const_string_proxy.h.

6.92.4.12 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator== (const char * *other*) [inline]

Definition at line 99 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:

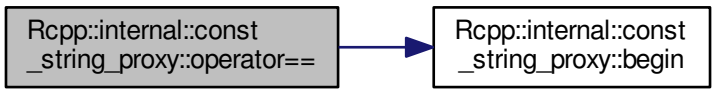


6.92.4.13 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator== (const const_string_proxy< RTYPE > & *other*) [inline]

Definition at line 106 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:



6.92.4.14 template<int RTYPE> bool Rcpp::internal::const_string_proxy< RTYPE >::operator== (SEXP *other*) const [inline]

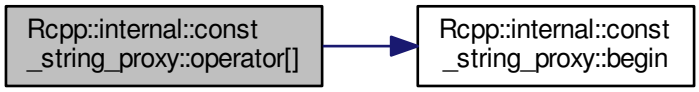
Definition at line 113 of file const_string_proxy.h.

6.92.4.15 template<int RTYPE> reference Rcpp::internal::const_string_proxy< RTYPE >::operator[] (R_xlen_t *n*) [inline]

Definition at line 97 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Here is the call graph for this function:



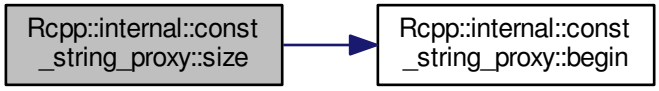
6.92.4.16 template<int RTYPE> R_xlen_t Rcpp::internal::const_string_proxy< RTYPE >::size ( ) const [inline]

Definition at line 95 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::begin().

Referenced by Rcpp::internal::const_string_proxy< RTYPE >::end().

Here is the call graph for this function:



6.92.5 Friends And Related Function Documentation

6.92.5.1 template<int RTYPE> template<int RT> std::string operator+ (const std::string & *x*, const const_string_proxy< RT > & *proxy*) [friend]

Referenced by Rcpp::internal::const_string_proxy< RTYPE >::operator char *().

6.92.5.2 template<int RTYPE> template<int RT> std::ostream& operator<< (std::ostream & *os*, const const_string_proxy< RT > & *proxy*) [friend]

Prints the element this proxy refers to to an output stream

6.92.6 Member Data Documentation

6.92.6.1 template<int RTYPE> std::string Rcpp::internal::const_string_proxy< RTYPE >::buffer [static], [private]

Definition at line 122 of file const_string_proxy.h.

6.92.6.2 template<int RTYPE> R_xlen_t Rcpp::internal::const_string_proxy< RTYPE >::index

Definition at line 86 of file const_string_proxy.h.

Referenced by Rcpp::internal::const_string_proxy< RTYPE >::import(), and Rcpp::internal::string_proxy< RTYPE >::import().

6.92.6.3 template<int RTYPE> const VECTOR* Rcpp::internal::const_string_proxy< RTYPE >::parent

Definition at line 85 of file const_string_proxy.h.

Referenced by Rcpp::internal::const_string_proxy< RTYPE >::const_string_proxy(), Rcpp::internal::const_string_proxy< RTYPE >::import(), and Rcpp::internal::string_proxy< RTYPE >::import().

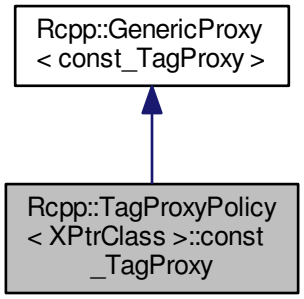
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/const_string_proxy.h

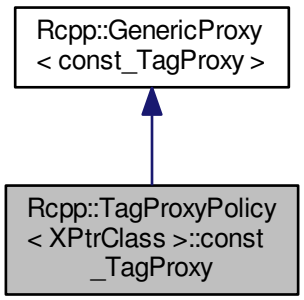
6.93 Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy Class Reference

`#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

6.93.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy

Definition at line 53 of file TagProxy.h.

6.93.2 Constructor & Destructor Documentation

6.93.2.1 template<typename XPtrClass> Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::const_TagProxy (XPtrClass & *xp_*) [inline]

Definition at line 55 of file TagProxy.h.

6.93.3 Member Function Documentation

6.93.3.1 template<typename XPtrClass> SEXP Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::get ( ) const [inline], [private]

Definition at line 66 of file TagProxy.h.

6.93.3.2 template<typename CLASS > Rcpp::TagProxyPolicy< CLASS >::const_TagProxy::operator SEXP ( ) const

Definition at line 122 of file proxy.h.

6.93.3.3 template<typename XPtrClass> template<typename U > Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::operator U ( ) const

6.93.4 Member Data Documentation

6.93.4.1 template<typename XPtrClass> XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::xp [private]

Definition at line 64 of file TagProxy.h.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/TagProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.94 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

6.94.1 Detailed Description

template<typename T>
class Rcpp::ConstInputParameter< T >

Definition at line 56 of file InputParameter.h.

6.94.2 Member Typedef Documentation

6.94.2.1 template<typename T > typedef const T Rcpp::ConstInputParameter< T >::const_nonref

Definition at line 58 of file InputParameter.h.

6.94.3 Constructor & Destructor Documentation

6.94.3.1 template<typename T > Rcpp::ConstInputParameter< T >::ConstInputParameter (SEXP *x_*) [inline]

Definition at line 59 of file InputParameter.h.

6.94.4 Member Function Documentation

6.94.4.1 template<typename T > Rcpp::ConstInputParameter< T >::operator const_nonref ( ) [inline]

Definition at line 61 of file InputParameter.h.

6.94.5 Member Data Documentation

6.94.5.1 template<typename T > T Rcpp::ConstInputParameter< T >::obj [private]

Definition at line 64 of file InputParameter.h.

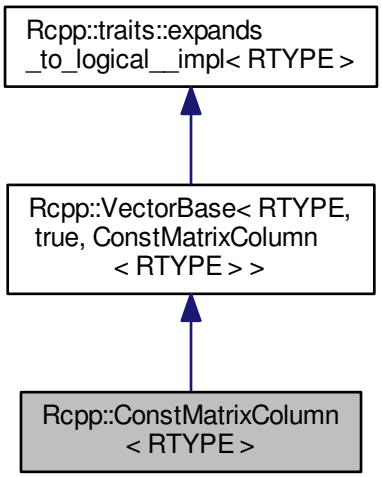
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

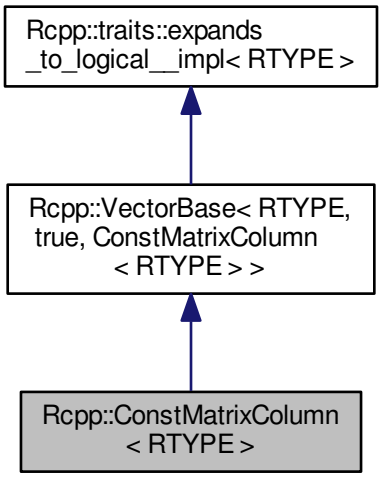
6.95 Rcpp::ConstMatrixColumn< RTYPE > Class Template Reference

`#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
- int size () const

Private Attributes

- const int n
- const_iterator const_start

6.95.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixColumn< RTYPE >

Definition at line 107 of file MatrixColumn.h.

6.95.2 Member Typedef Documentation

6.95.2.1 template<int RTYPE> typedef MATRIX::const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_iterator

Definition at line 112 of file MatrixColumn.h.

6.95.2.2 template<int RTYPE> typedef MATRIX::const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::const_Proxy

Definition at line 110 of file MatrixColumn.h.

6.95.2.3 template<int RTYPE> typedef Matrix<RTYPE> Rcpp::ConstMatrixColumn< RTYPE >::MATRIX

Definition at line 109 of file MatrixColumn.h.

6.95.2.4 template<int RTYPE> typedef MATRIX::value_type Rcpp::ConstMatrixColumn< RTYPE >::value_type

Definition at line 111 of file MatrixColumn.h.

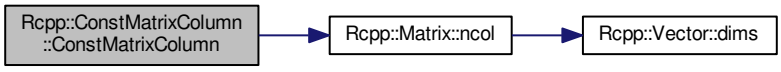
6.95.3 Constructor & Destructor Documentation

6.95.3.1 template<int RTYPE> Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (const MATRIX & *parent*, int *i*) [inline]

Definition at line 114 of file MatrixColumn.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:



6.95.3.2 template<int RTYPE> Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (const ConstMatrixColumn< RTYPE > & *other*) [inline]

Definition at line 121 of file MatrixColumn.h.

6.95.4 Member Function Documentation

6.95.4.1 template<int RTYPE> const_iterator Rcpp::ConstMatrixColumn< RTYPE >::begin ( ) const [inline]

Definition at line 129 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.95.4.2 template<int RTYPE> const_iterator Rcpp::ConstMatrixColumn< RTYPE >::end ( ) const [inline]

Definition at line 133 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start, and Rcpp::MatrixColumn< RTYPE >::n.

6.95.4.3 template<int RTYPE> const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::operator[ ] (int *i*) const [inline]

Definition at line 125 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.95.4.4 template<int RTYPE> int Rcpp::ConstMatrixColumn< RTYPE >::size ( ) const [inline]

Definition at line 137 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n.

6.95.5 Member Data Documentation

6.95.5.1 template<int RTYPE> const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_start [private]

Definition at line 143 of file MatrixColumn.h.

6.95.5.2 template<int RTYPE> const int Rcpp::ConstMatrixColumn< RTYPE >::n [private]

Definition at line 142 of file MatrixColumn.h.

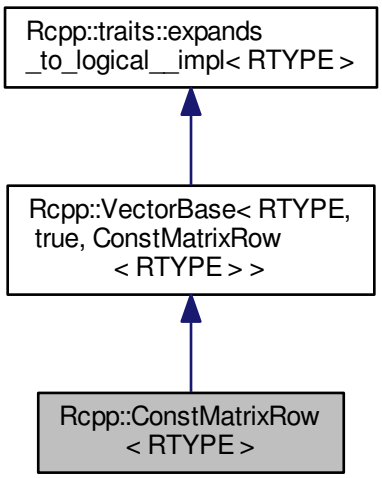
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

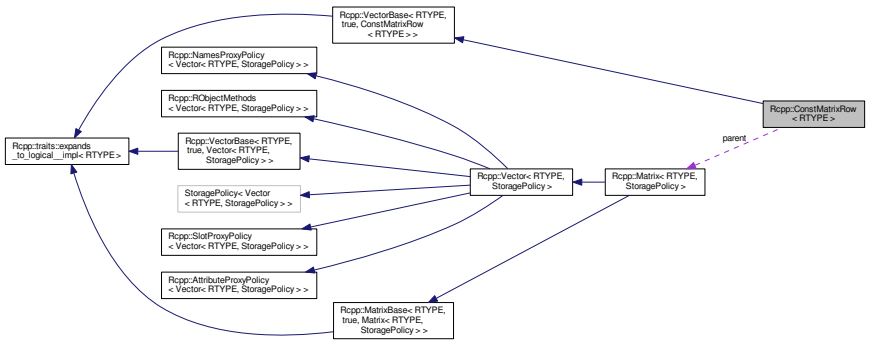
6.96 Rcpp::ConstMatrixRow< RTYPE > Class Template Reference

`#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

6.96.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >

Definition at line 175 of file MatrixRow.h.

6.96.2 Member Typedef Documentation

6.96.2.1 template<int RTYPE> typedef MATRIX::const_Proxy Rcpp::ConstMatrixRow< RTYPE >::const_reference

Definition at line 178 of file MatrixRow.h.

6.96.2.2 template<int RTYPE> typedef const_iterator Rcpp::ConstMatrixRow< RTYPE >::iterator

Definition at line 249 of file MatrixRow.h.

6.96.2.3 template<int RTYPE> typedef Matrix<RTYPE> Rcpp::ConstMatrixRow< RTYPE >::MATRIX

Definition at line 177 of file MatrixRow.h.

6.96.2.4 template<int RTYPE> typedef MATRIX::value_type Rcpp::ConstMatrixRow< RTYPE >::value_type

Definition at line 179 of file MatrixRow.h.

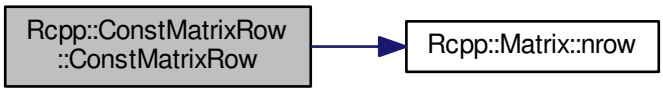
6.96.3 Constructor & Destructor Documentation

6.96.3.1 template<int RTYPE> Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (const MATRIX & *object*, int *i*) [inline]

Definition at line 251 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::MatrixRow< RTYPE >::parent.

Here is the call graph for this function:



6.96.3.2 template<int RTYPE> Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (const ConstMatrixRow< RTYPE > & *other*) [inline]

Definition at line 260 of file MatrixRow.h.

6.96.4 Member Function Documentation

6.96.4.1 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::begin ( ) const [inline]

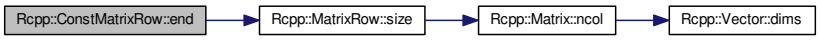
Definition at line 271 of file MatrixRow.h.

6.96.4.2 template<int RTYPE> const_iterator Rcpp::ConstMatrixRow< RTYPE >::end ( ) const [inline]

Definition at line 275 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



6.96.4.3 template<int RTYPE> int Rcpp::ConstMatrixRow< RTYPE >::get_parent_index (int *i*) const [inline], [private]

Definition at line 289 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent_nrow, and RCPP_DEBUG_4.

6.96.4.4 template<int RTYPE> const_reference Rcpp::ConstMatrixRow< RTYPE >::operator[ ] (int *i*) const [inline]

Definition at line 267 of file MatrixRow.h.

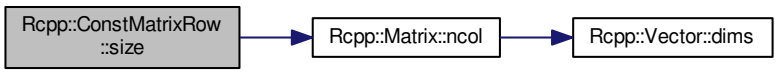
References Rcpp::MatrixRow< RTYPE >::parent, Rcpp::MatrixRow< RTYPE >::parent_nrow, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.96.4.5 template<int RTYPE> int Rcpp::ConstMatrixRow< RTYPE >::size ( ) const [inline]

Definition at line 279 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::MatrixRow< RTYPE >::parent.

Here is the call graph for this function:



6.96.5 Member Data Documentation

6.96.5.1 template<int RTYPE> const MATRIX& Rcpp::ConstMatrixRow< RTYPE >::parent [private]

Definition at line 284 of file MatrixRow.h.

6.96.5.2 template<int RTYPE> int Rcpp::ConstMatrixRow< RTYPE >::parent_nrow [private]

Definition at line 286 of file MatrixRow.h.

6.96.5.3 template<int RTYPE> int Rcpp::ConstMatrixRow< RTYPE >::row [private]

Definition at line 287 of file MatrixRow.h.

6.96.5.4 template<int RTYPE> MATRIX::const_iterator Rcpp::ConstMatrixRow< RTYPE >::start [private]

Definition at line 285 of file MatrixRow.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.97 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

6.97.1 Detailed Description

template<typename T>
class Rcpp::ConstReferenceInputParameter< T >

Definition at line 69 of file InputParameter.h.

6.97.2 Member Typedef Documentation

6.97.2.1 template<typename T > typedef const T& Rcpp::ConstReferenceInputParameter< T >::const_reference

Definition at line 71 of file InputParameter.h.

6.97.3 Constructor & Destructor Documentation

6.97.3.1 template<typename T > Rcpp::ConstReferenceInputParameter< T >::ConstReferenceInputParameter (SEXP *x_*) [inline]

Definition at line 72 of file InputParameter.h.

6.97.4 Member Function Documentation

6.97.4.1 template<typename T > Rcpp::ConstReferenceInputParameter< T >::operator const_reference ( ) [inline]

Definition at line 74 of file InputParameter.h.

6.97.5 Member Data Documentation

6.97.5.1 template<typename T > T Rcpp::ConstReferenceInputParameter< T >::obj [private]

Definition at line 77 of file InputParameter.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.98 Rcpp::traits::container_exporter< Container, double > Struct Template Reference

```
#include <export.h>
```

Public Types

- typedef ContainerExporter< Container, double > type

6.98.1 Detailed Description

template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, double >

Definition at line 65 of file export.h.

6.98.2 Member Typedef Documentation

6.98.2.1 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

6.99 Rcpp::traits::container_exporter< Container, int > Struct Template Reference

```
#include <export.h>
```

Public Types

- typedef ContainerExporter< Container, int > type

6.99.1 Detailed Description

template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, int >

Definition at line 62 of file export.h.

6.99.2 Member Typedef Documentation

6.99.2.1 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.

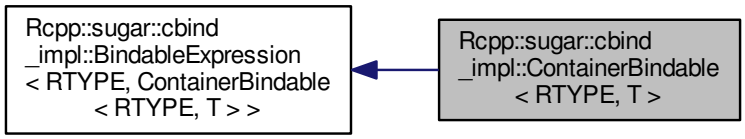
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

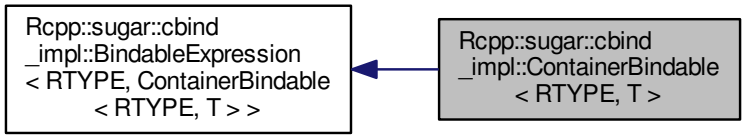
6.100 Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T > Class Template Reference

`#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

6.100.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >

Definition at line 82 of file cbind.h.

6.100.2 Member Typedef Documentation

6.100.2.1 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.

6.100.3 Constructor & Destructor Documentation

6.100.3.1 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.

6.100.3.2 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.

6.100.3.3 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.

6.100.4 Member Function Documentation

6.100.4.1 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.

6.100.4.2 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.

6.100.4.3 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.

6.100.4.4 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.

6.100.4.5 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.

6.100.5 Member Data Documentation

6.100.5.1 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.

6.100.5.2 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.

6.100.5.3 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.

6.100.5.4 template<int RTYPE, typename T > T Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec [private]

Definition at line 88 of file cbind.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.101 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

6.101.1 Detailed Description

template<template< class, class > class ContainerTemplate, typename T>
class Rcpp::traits::ContainerExporter< ContainerTemplate, T >

Definition at line 41 of file export.h.

6.101.2 Member Typedef Documentation

6.101.2.1 template<template< class, class > class ContainerTemplate, typename T > typedef ContainerTemplate<T, std::allocator<T> > Rcpp::traits::ContainerExporter< ContainerTemplate, T >::Container

Definition at line 43 of file export.h.

6.101.3 Constructor & Destructor Documentation

6.101.3.1 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.

6.101.3.2 template<template< class, class > class ContainerTemplate, typename T > Rcpp::traits::ContainerExporter< ContainerTemplate, T >::~ContainerExporter ( ) [inline]

Definition at line 47 of file export.h.

6.101.4 Member Function Documentation

6.101.4.1 template<template< class, class > class ContainerTemplate, typename T > Container Rcpp::traits::ContainerExporter< ContainerTemplate, T >::get ( ) [inline]

Definition at line 49 of file export.h.

6.101.5 Member Data Documentation

6.101.5.1 template<template< class, class > class ContainerTemplate, typename T > SEXP Rcpp::traits::ContainerExporter< ContainerTemplate, T >::object [private]

Definition at line 60 of file export.h.

6.101.5.2 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.

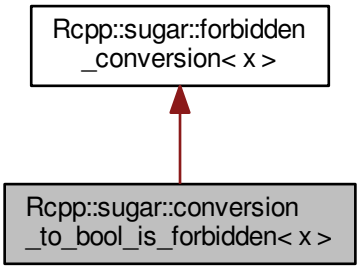
The documentation for this class was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

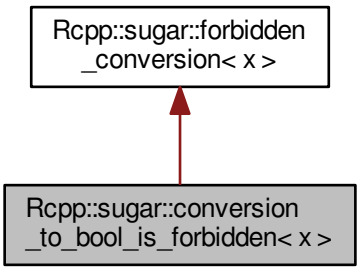
6.102 Rcpp::sugar::conversion_to_bool_is_forbidden< x > Class Template Reference

`#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 ()

6.102.1 Detailed Description

template<bool x>
class Rcpp::sugar::conversion_to_bool_is_forbidden< x >

Definition at line 35 of file SingleLogicalResult.h.

6.102.2 Member Function Documentation

6.102.2.1 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< false, Any< false, T > >::operator bool().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

6.103 tinyformat::detail::convertToInt< T, convertible > Struct Template Reference

`#include <tinyformat.h>`

Static Public Member Functions

- static int invoke (const T &)

6.103.1 Detailed Description

template<typename T, bool convertible = is_convertible<T,int>::value>
struct tinyformat::detail::convertToInt< T, convertible >

Definition at line 262 of file tinyformat.h.

6.103.2 Member Function Documentation

6.103.2.1 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 264 of file tinyformat.h.

References TINYFORMAT_ERROR.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.104 tinyformat::detail::convertToInt< T, true > Struct Template Reference

`#include <tinyformat.h>`

Static Public Member Functions

- static int invoke (const T &value)

6.104.1 Detailed Description

template<typename T>
struct tinyformat::detail::convertToInt< T, true >

Definition at line 273 of file tinyformat.h.

6.104.2 Member Function Documentation

6.104.2.1 template<typename T > static int tinyformat::detail::convertToInt< T, true >::invoke (const T & *value*) [inline], [static]

Definition at line 275 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.105 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

6.105.1 Detailed Description

template<typename HASH, typename STORAGE>
class Rcpp::sugar::CountInserter< HASH, STORAGE >

Definition at line 29 of file table.h.

6.105.2 Constructor & Destructor Documentation

6.105.2.1 template<typename HASH , typename STORAGE > Rcpp::sugar::CountInserter< HASH, STORAGE >::CountInserter (HASH & *hash_*) [inline]

Definition at line 31 of file table.h.

6.105.3 Member Function Documentation

6.105.3.1 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.

6.105.4 Member Data Documentation

6.105.4.1 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()(), and Rcpp::sugar::Table< RTYPE, T↩ ABLE_T >::Table().

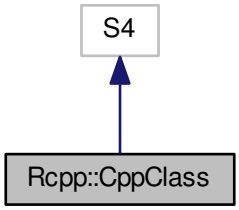
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

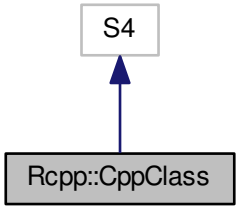
6.106 Rcpp::CppClass Class Reference

`#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

6.106.1 Detailed Description

Definition at line 382 of file Module.h.

6.106.2 Member Typedef Documentation

6.106.2.1 typedef S4 Rcpp::CppClass::Base [private]

Definition at line 383 of file Module.h.

6.106.2.2 typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppClass::XP

Definition at line 386 of file Module.h.

6.106.2.3 typedef XPtr<class_Base> Rcpp::CppClass::XP_Class

Definition at line 385 of file Module.h.

6.106.3 Constructor & Destructor Documentation

6.106.3.1 Rcpp::CppClass::CppClass (SEXP *x*) [inline]

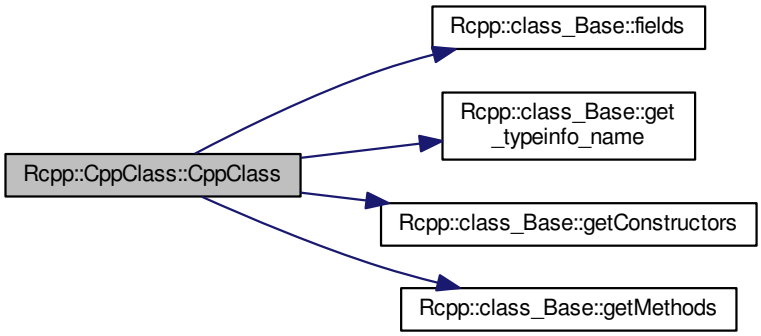
Definition at line 387 of file Module.h.

6.106.3.2 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, Rcpp::class_Base::parents, and RCPP_CTOR_ASSIGN_WITH_BASE.

Here is the call graph for this function:

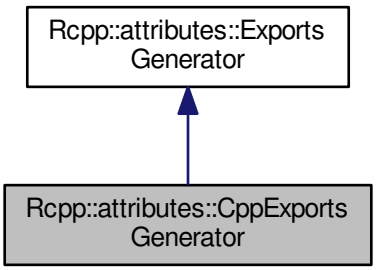


The documentation for this class was generated from the following file:

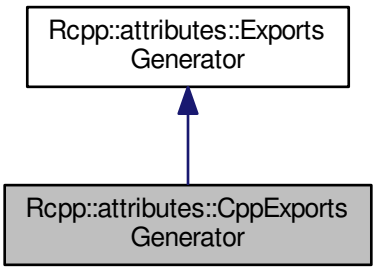
- inst/include/Rcpp/Module.h

6.107 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 ()
- 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 > cppExports_

Additional Inherited Members

6.107.1 Detailed Description

Definition at line 644 of file attributes.cpp.

6.107.2 Constructor & Destructor Documentation

6.107.2.1 Rcpp::attributes::CppExportsGenerator::CppExportsGenerator (const std::string & *packageDir,* const std::string & *package,* const std::string & *fileSep*) [explicit]

Definition at line 1807 of file attributes.cpp.

6.107.3 Member Function Documentation

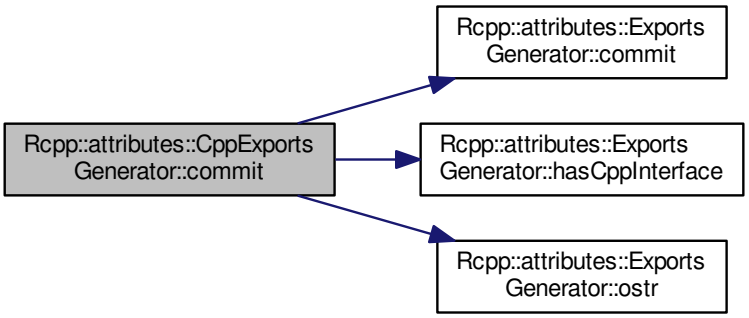
6.107.3.1 bool Rcpp::attributes::CppExportsGenerator::commit (const std::vector< std::string > & *includes*) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 1920 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), and Rcpp::attributes::ExportsGenerator::ostr().

Here is the call graph for this function:



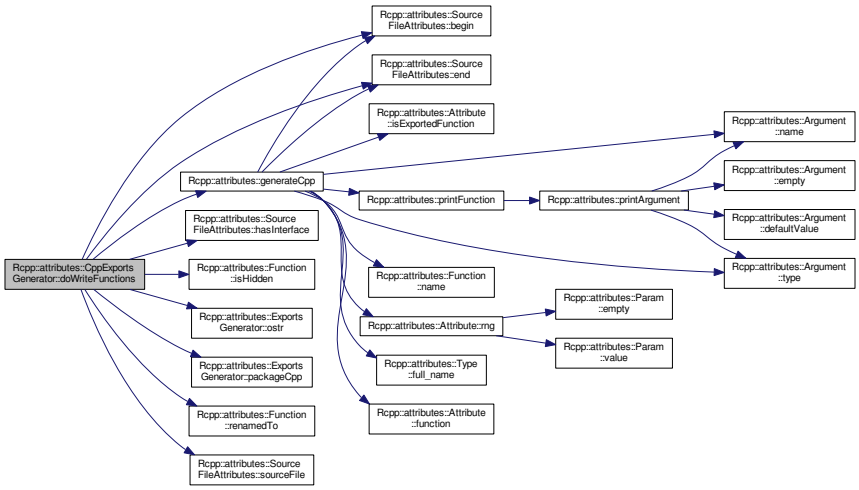
6.107.3.2 void Rcpp::attributes::CppExportsGenerator::doWriteFunctions (const SourceFileAttributes & *attributes*, bool *verbose*) [private], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 1817 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), cppExports_, Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::generateCpp(), Rcpp::attributes::SourceFileAttributes::hasInterface(), Rcpp::attributes::Function::isHidden(), Rcpp::attributes::kInterfaceCpp, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::packageCpp(), Rcpp::Rcout, Rcpp::attributes::Function::renamedTo(), and Rcpp::attributes::SourceFileAttributes::sourceFile().

Here is the call graph for this function:



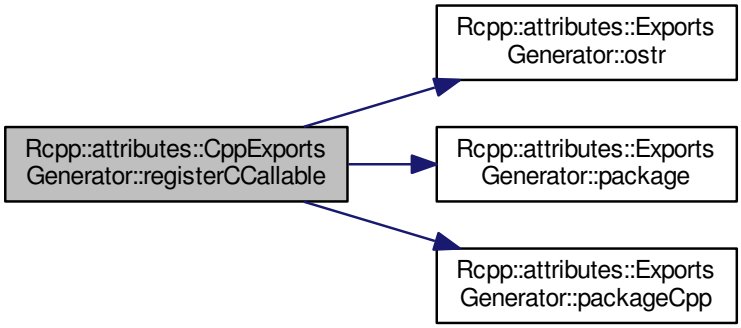
6.107.3.3 std::string Rcpp::attributes::CppExportsGenerator::registerCCallable (size_t *indent*, const std::string & *exportedName*, const std::string & *name*) const [private]

Definition at line 1908 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), and Rcpp↩ ::attributes::ExportsGenerator::packageCpp().

Referenced by writeEnd().

Here is the call graph for this function:



6.107.3.4 virtual void Rcpp::attributes::CppExportsGenerator::writeBegin ( ) [inline], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 650 of file attributes.cpp.

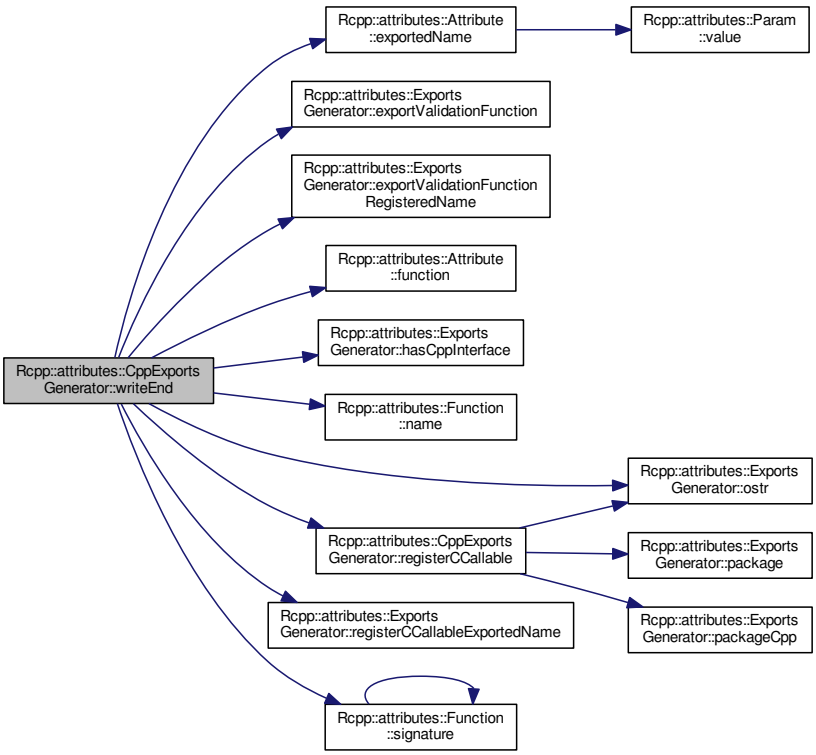
6.107.3.5 void Rcpp::attributes::CppExportsGenerator::writeEnd ( ) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 1855 of file attributes.cpp.

References cppExports_, Rcpp::attributes::Attribute::exportedName(), Rcpp::attributes::ExportsGenerator::exportValidationFunction(), Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), Rcpp::attributes::Function::name(), Rcpp::attributes::ExportsGenerator::ostr(), registerCCallable(), Rcpp::attributes::ExportsGenerator::registerCCallableExportedName(), and Rcpp::attributes::Function::signature().

Here is the call graph for this function:



6.107.4 Member Data Documentation

6.107.4.1 std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::cppExports_ [private]

Definition at line 663 of file attributes.cpp.

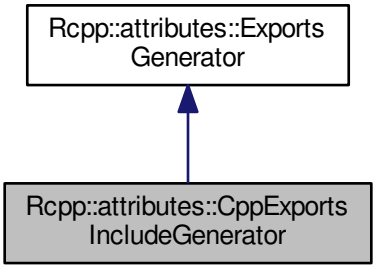
Referenced by doWriteFunctions(), and writeEnd().

The documentation for this class was generated from the following file:

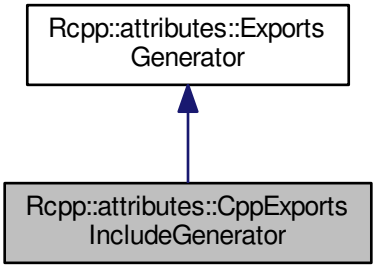
- src/attributes.cpp

6.108 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 ()
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)
- std::string getCCallable (const std::string &function) const
- std::string getHeaderGuard () const

Private Attributes

- std::string includeDir_

Additional Inherited Members

6.108.1 Detailed Description

Definition at line 667 of file attributes.cpp.

6.108.2 Constructor & Destructor Documentation

6.108.2.1 Rcpp::attributes::CppExportsIncludeGenerator::CppExportsIncludeGenerator (const std::string & *packageDir,* const std::string & *package,* const std::string & *fileSep*)

Definition at line 1941 of file attributes.cpp.

References includeDir_.

6.108.3 Member Function Documentation

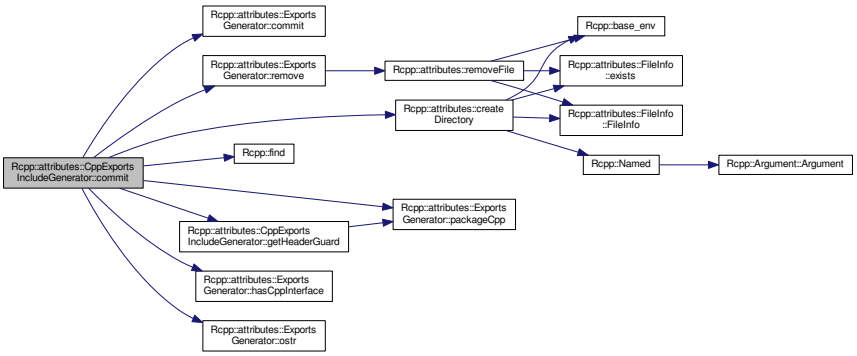
6.108.3.1 bool Rcpp::attributes::CppExportsIncludeGenerator::commit (const std::vector< std::string > & *includes*) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2090 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:



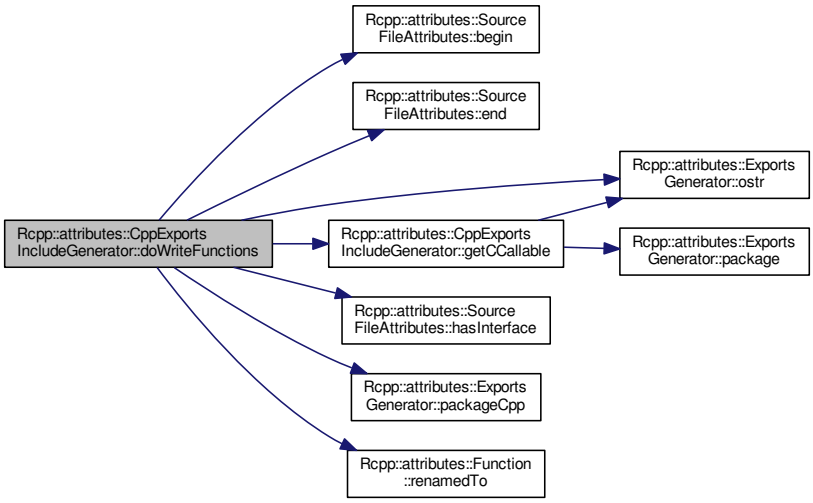
6.108.3.2 void Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions (const SourceFileAttributes & *attributes,* bool *verbose*) [private], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2003 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes::end(), getCallable(), Rcpp::attributes::SourceFileAttributes::hasInterface(), Rcpp::attributes::kInterfaceCpp, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::packageCpp(), and Rcpp::attributes::Function::renamedTo().

Here is the call graph for this function:



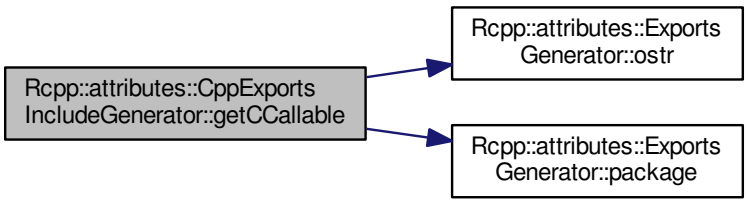
6.108.3.3 std::string Rcpp::attributes::CppExportsIncludeGenerator::getCCallable (const std::string & *function*) const [private]

Definition at line 2142 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:



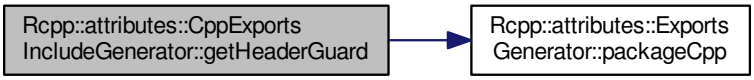
6.108.3.4 std::string Rcpp::attributes::CppExportsIncludeGenerator::getHeaderGuard ( ) const [private]

Definition at line 2151 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::packageCpp().

Referenced by commit(), and writeEnd().

Here is the call graph for this function:



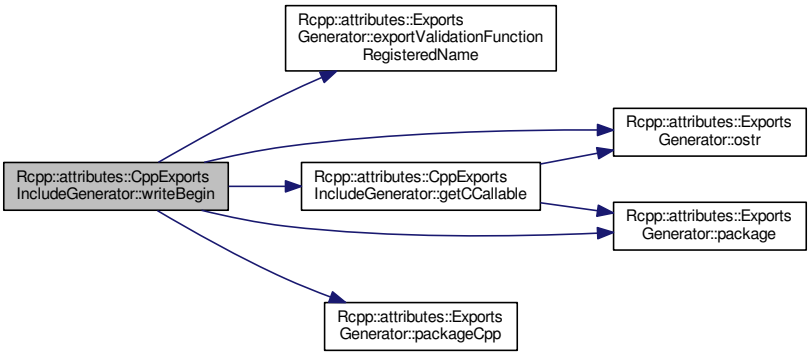
6.108.3.5 void Rcpp::attributes::CppExportsIncludeGenerator::writeBegin ( ) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 1954 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName(), getCCallable(), Rcpp↩
::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), and Rcpp::attributes::Exports↩
Generator::packageCpp().

Here is the call graph for this function:



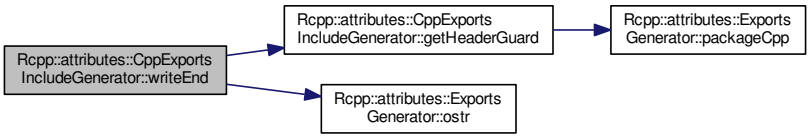
6.108.3.6 void Rcpp::attributes::CppExportsIncludeGenerator::writeEnd ( ) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2084 of file attributes.cpp.

References getHeaderGuard(), and Rcpp::attributes::ExportsGenerator::ostr().

Here is the call graph for this function:



6.108.4 Member Data Documentation

6.108.4.1 std::string Rcpp::attributes::CppExportsIncludeGenerator::includeDir_ [private]

Definition at line 684 of file attributes.cpp.

Referenced by commit(), and CppExportsIncludeGenerator().

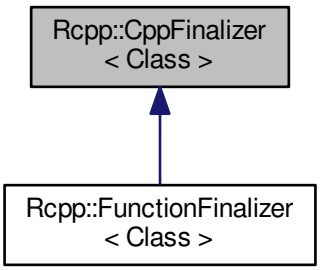
The documentation for this class was generated from the following file:

- src/attributes.cpp

6.109 Rcpp::CppFinalizer< Class > Class Template Reference

`#include <Module.h>`

Inheritance diagram for Rcpp::CppFinalizer< Class >:



Public Member Functions

- CppFinalizer ()
- virtual void run (Class ∗)

6.109.1 Detailed Description

template<typename Class>
class Rcpp::CppFinalizer< Class >

Definition at line 300 of file Module.h.

6.109.2 Constructor & Destructor Documentation

6.109.2.1 template<typename Class > Rcpp::CppFinalizer< Class >::CppFinalizer ( ) [inline]

Definition at line 302 of file Module.h.

6.109.3 Member Function Documentation

6.109.3.1 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.

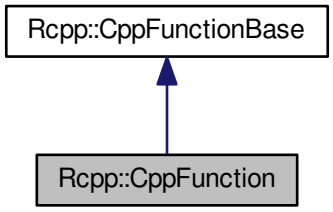
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

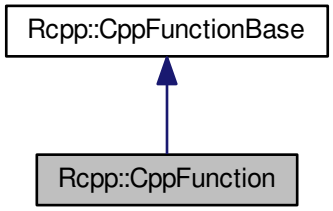
6.110 Rcpp::CppFunction Class Reference

`#include <CppFunction.h>`

Inheritance diagram for Rcpp::CppFunction:



Collaboration diagram for Rcpp::CppFunction:



Public Member Functions

- CppFunction (const char *doc=0)
- virtual ~CppFunction ()
- 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

6.110.1 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 CppFunction (see Module_generated_CppFunction.h for all the definitions

Definition at line 52 of file CppFunction.h.

6.110.2 Constructor & Destructor Documentation

6.110.2.1 Rcpp::CppFunction::CppFunction (const char * *doc* = 0) [inline]

Definition at line 54 of file CppFunction.h.

6.110.2.2 virtual Rcpp::CppFunction::~CppFunction ( ) [inline], [virtual]

Definition at line 55 of file CppFunction.h.

6.110.3 Member Function Documentation

6.110.3.1 virtual SEXP Rcpp::CppFunction::get_formals ( ) [inline], [virtual]

formal arguments

Definition at line 75 of file CppFunction.h.

Referenced by Rcpp::Module::get_function().

6.110.3.2 virtual DL_FUNC Rcpp::CppFunction::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().

6.110.3.3 virtual bool Rcpp::CppFunction::is_void ( ) [inline], [virtual]

voidness

Definition at line 65 of file CppFunction.h.

Referenced by Rcpp::Module::get_function(), and Rcpp::Module::invoke().

6.110.3.4 virtual int Rcpp::CppFunction::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().

6.110.3.5 virtual void Rcpp::CppFunction::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().

6.110.4 Member Data Documentation

6.110.4.1 std::string Rcpp::CppFunction::docstring

description of the function

Definition at line 86 of file CppFunction.h.

Referenced by Rcpp::Module::get_function().

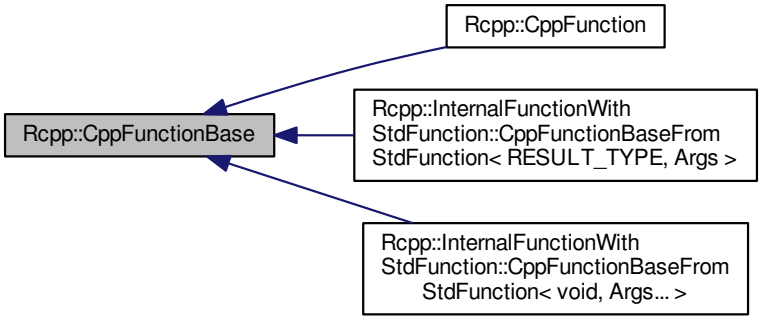
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/CppFunction.h

6.111 Rcpp::CppFunctionBase Class Reference

`#include <CppFunction.h>`

Inheritance diagram for Rcpp::CppFunctionBase:



Public Member Functions

- CppFunctionBase ()
- virtual ~CppFunctionBase ()
- virtual SEXP operator() (SEXP *)

6.111.1 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.

6.111.2 Constructor & Destructor Documentation

6.111.2.1 Rcpp::CppFunctionBase::CppFunctionBase ( ) [inline]

Definition at line 33 of file CppFunction.h.

6.111.2.2 virtual Rcpp::CppFunctionBase::~CppFunctionBase ( ) [inline], [virtual]

Definition at line 34 of file CppFunction.h.

6.111.3 Member Function Documentation

6.111.3.1 virtual SEXP Rcpp::CppFunctionBase::operator() (SEXP *) [inline], [virtual]

modules call the function with this interface. input: an array of SEXP output: a SEXP.

Reimplemented in Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >, and Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >.

Definition at line 40 of file CppFunction.h.

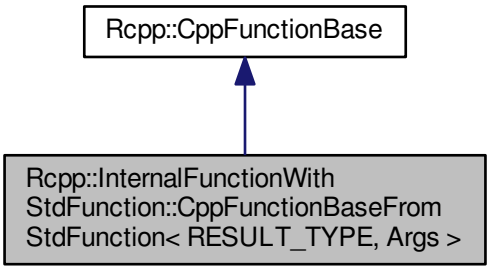
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/CppFunction.h

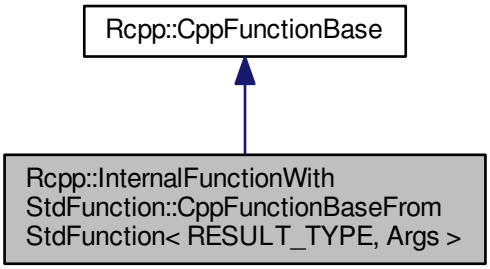
6.112 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args > Class Template Reference

`#include <InternalFunctionWithStdFunction.h>`

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >:



Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >:



Public Member Functions

- CppFunctionBaseFromStdFunction (const std::function< RESULT_TYPE(Args...)> &fun)
- virtual ~CppFunctionBaseFromStdFunction ()
- SEXP operator() (SEXP *args)

Private Attributes

- const std::function< RESULT_TYPE(Args...)> fun

6.112.1 Detailed Description

template<typename RESULT_TYPE, typename... Args>
class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >

Definition at line 35 of file InternalFunctionWithStdFunction.h.

6.112.2 Constructor & Destructor Documentation

6.112.2.1 template<typename RESULT_TYPE , typename... Args> Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::CppFunctionBaseFromStdFunction (const std::function< RESULT_TYPE(Args...)> & *fun*) [inline]

Definition at line 37 of file InternalFunctionWithStdFunction.h.

6.112.2.2 template<typename RESULT_TYPE , typename... Args> virtual Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::~CppFunctionBaseFromStdFunction ( ) [inline], [virtual]

Definition at line 38 of file InternalFunctionWithStdFunction.h.

6.112.3 Member Function Documentation

6.112.3.1 template<typename RESULT_TYPE , typename... Args> SEXP Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< 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::CppFunctionBase.

Definition at line 40 of file InternalFunctionWithStdFunction.h.

References BEGIN_RCPP, END_RCPP, and Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStd↩Function< RESULT_TYPE, Args >::fun.

6.112.4 Member Data Documentation

6.112.4.1 template<typename RESULT_TYPE , typename... Args> const std::function<RESULT_TYPE(Args...)> Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::fun [private]

Definition at line 48 of file InternalFunctionWithStdFunction.h.

Referenced by Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::operator()(), and Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::operator()().

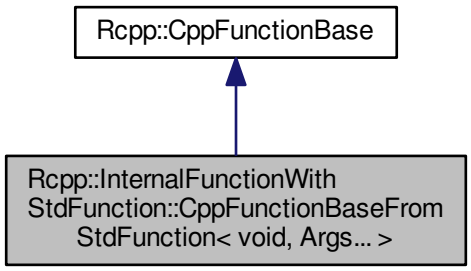
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InternalFunctionWithStdFunction.h

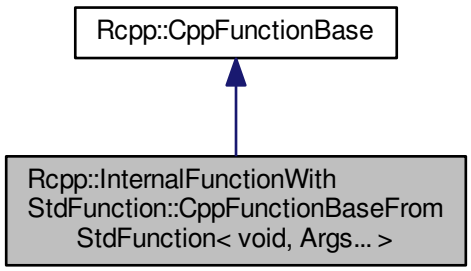
6.113 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... > Class Template Reference

`#include <InternalFunctionWithStdFunction.h>`

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >:



Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >:



Public Member Functions

- CppFunctionBaseFromStdFunction (const std::function< void(Args...)> &fun)
- virtual ~CppFunctionBaseFromStdFunction ()
- SEXP operator() (SEXP *args)

Private Attributes

- const std::function< void(Args...)> fun

6.113.1 Detailed Description

template<typename... Args>
class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >

Definition at line 52 of file InternalFunctionWithStdFunction.h.

6.113.2 Constructor & Destructor Documentation

6.113.2.1 template<typename... Args> Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::CppFunctionBaseFromStdFunction (const std::function< void(Args...)> & *fun*) [inline]

Definition at line 54 of file InternalFunctionWithStdFunction.h.

6.113.2.2 template<typename... Args> virtual Rcpp::InternalFunctionWithStdFunction::CppFunctionBase↩ FromStdFunction< void, Args... >::~CppFunctionBaseFromStdFunction ( ) [inline], [virtual]

Definition at line 55 of file InternalFunctionWithStdFunction.h.

6.113.3 Member Function Documentation

6.113.3.1 template<typename... Args> SEXP Rcpp::InternalFunctionWithStdFunction::Cpp↩ FunctionBaseFromStdFunction< void, Args... >::operator() (SEXP *) [inline], [virtual]

modules call the function with this interface. input: an array of SEXP output: a SEXP.

Reimplemented from Rcpp::CppFunctionBase.

Definition at line 57 of file InternalFunctionWithStdFunction.h.

References BEGIN_RCPP, END_RCPP, and Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStd↩ Function< RESULT_TYPE, Args >::fun.

6.113.4 Member Data Documentation

6.113.4.1 template<typename... Args> const std::function<void(Args...)> Rcpp::InternalFunctionWithStdFunction::↩ CppFunctionBaseFromStdFunction< void, Args... >::fun [private]

Definition at line 64 of file InternalFunctionWithStdFunction.h.

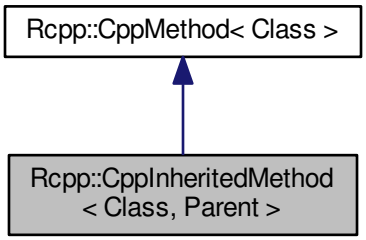
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InternalFunctionWithStdFunction.h

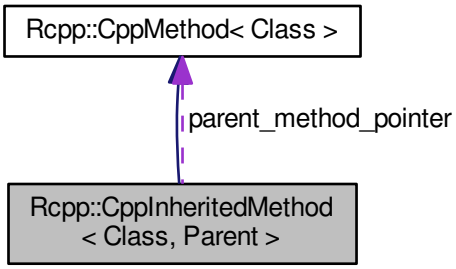
6.114 Rcpp::CppInheritedMethod< Class, Parent > Class Template Reference

`#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

6.114.1 Detailed Description

template<typename Class, typename Parent>
class Rcpp::CppInheritedMethod< Class, Parent >

Definition at line 112 of file Module.h.

6.114.2 Member Typedef Documentation

6.114.2.1 template<typename Class , typename Parent > typedef CppMethod<Parent> Rcpp::CppInheritedMethod< Class, Parent >::ParentMethod

Definition at line 115 of file Module.h.

6.114.2.2 template<typename Class , typename Parent > typedef Rcpp::XPtr<Class> Rcpp::CppInheritedMethod< Class, Parent >::XP

Definition at line 114 of file Module.h.

6.114.3 Constructor & Destructor Documentation

6.114.3.1 template<typename Class , typename Parent > Rcpp::CppInheritedMethod< Class, Parent >::CppInheritedMethod (ParentMethod * *parent_method_pointer_*) [inline]

Definition at line 117 of file Module.h.

6.114.4 Member Function Documentation

6.114.4.1 template<typename Class , typename Parent > bool Rcpp::CppInheritedMethod< Class, Parent >::is_const ( ) [inline], [virtual]

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 126 of file Module.h.

6.114.4.2 template<typename Class , typename Parent > bool Rcpp::CppInheritedMethod< Class, Parent >::is_void ( ) [inline], [virtual]

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 125 of file Module.h.

6.114.4.3 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.

6.114.4.4 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.

6.114.4.5 template<typename Class , typename Parent > void Rcpp::CppInheritedMethod< Class, Parent >::signature (std::string & *s,* const char * *name*) [inline], [virtual]

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 127 of file Module.h.

6.114.5 Member Data Documentation

6.114.5.1 template<typename Class , typename Parent > ParentMethod* Rcpp::CppInheritedMethod< Class, Parent >::parent_method_pointer [private]

Definition at line 130 of file Module.h.

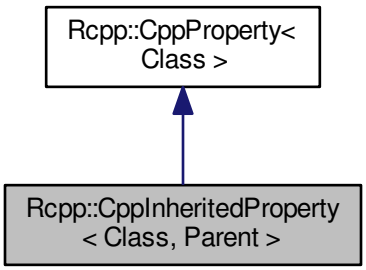
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

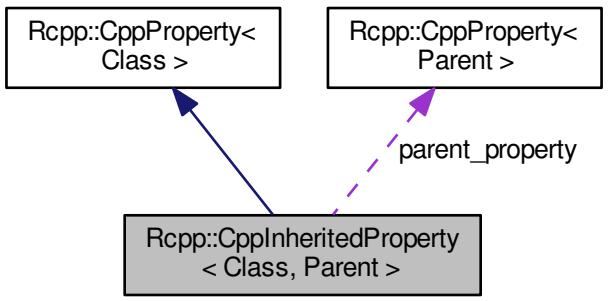
6.115 Rcpp::CppInheritedProperty< Class, Parent > Class Template Reference

`#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

6.115.1 Detailed Description

template<typename Class, typename Parent>
class Rcpp::CppInheritedProperty< Class, Parent >

Definition at line 281 of file Module.h.

6.115.2 Member Typedef Documentation

6.115.2.1 template<typename Class , typename Parent > typedef CppProperty<Class> Rcpp::CppInheritedProperty< Class, Parent >::Base

Definition at line 283 of file Module.h.

6.115.3 Constructor & Destructor Documentation

6.115.3.1 template<typename Class , typename Parent > Rcpp::CppInheritedProperty< Class, Parent >::CppInheritedProperty (CppProperty< Parent > * *parent_property_*) [inline]

Definition at line 285 of file Module.h.

6.115.4 Member Function Documentation

6.115.4.1 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.

6.115.4.2 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.

6.115.4.3 template<typename Class , typename Parent > bool Rcpp::CppInheritedProperty< Class, Parent >::is_readonly ( ) [inline], [virtual]

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 292 of file Module.h.

6.115.4.4 template<typename Class , typename Parent > void Rcpp::CppInheritedProperty< Class, Parent >::set (Class * *obj,* SEXP *s*) [inline], [virtual]

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 291 of file Module.h.

6.115.5 Member Data Documentation

6.115.5.1 template<typename Class , typename Parent > CppProperty<Parent>* Rcpp::CppInheritedProperty< Class, Parent >::parent_property [private]

Definition at line 296 of file Module.h.

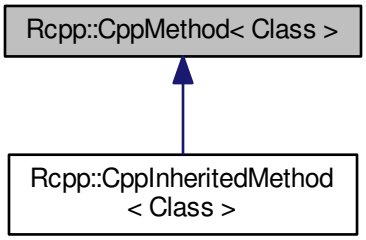
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.116 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 *object, SEXP *args)
- virtual ~CppMethod ()
- virtual int nargs ()
- virtual bool is_void ()
- virtual bool is_const ()
- virtual void signature (std::string &s, const char *name)

6.116.1 Detailed Description

template<typename Class>
class Rcpp::CppMethod< Class >

Definition at line 98 of file Module.h.

6.116.2 Member Typedef Documentation

6.116.2.1 template<typename Class > typedef Rcpp::XPtr<Class> Rcpp::CppMethod< Class >::XP

Definition at line 100 of file Module.h.

6.116.3 Constructor & Destructor Documentation

6.116.3.1 template<typename Class > Rcpp::CppMethod< Class >::CppMethod ( ) [inline]

Definition at line 102 of file Module.h.

6.116.3.2 template<typename Class > virtual Rcpp::CppMethod< Class >::~CppMethod ( ) [inline], [virtual]

Definition at line 104 of file Module.h.

6.116.4 Member Function Documentation

6.116.4.1 template<typename Class > virtual bool Rcpp::CppMethod< Class >::is_const ( ) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 107 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::is_const().

6.116.4.2 template<typename Class > virtual bool Rcpp::CppMethod< Class >::is_void ( ) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 106 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::is_void().

6.116.4.3 template<typename Class > virtual int Rcpp::CppMethod< Class >::nargs ( ) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 105 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::nargs().

6.116.4.4 template<typename Class > virtual SEXP Rcpp::CppMethod< Class >::operator() (Class * *object*, SEXP * *args*) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 103 of file Module.h.

6.116.4.5 template<typename Class > virtual void Rcpp::CppMethod< Class >::signature (std::string & *s*, const char * *name*) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 108 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::signature().

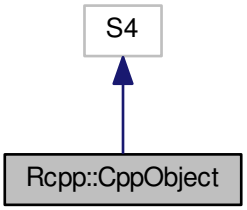
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

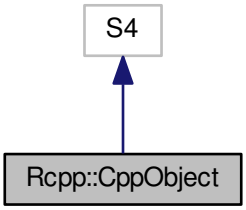
6.117 Rcpp::CppObject Class Reference

`#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

6.117.1 Detailed Description

Definition at line 412 of file Module.h.

6.117.2 Member Typedef Documentation

6.117.2.1 typedef S4 Rcpp::CppObject::Base [private]

Definition at line 413 of file Module.h.

6.117.2.2 typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppObject::XP

Definition at line 415 of file Module.h.

6.117.3 Constructor & Destructor Documentation

6.117.3.1 Rcpp::CppObject::CppObject (Module * *p*, class_Base * *clazz*, SEXP *xp*) [inline]

Definition at line 416 of file Module.h.

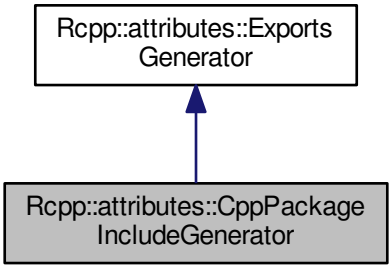
References RCPP_CTOR_ASSIGN_WITH_BASE.

The documentation for this class was generated from the following file:

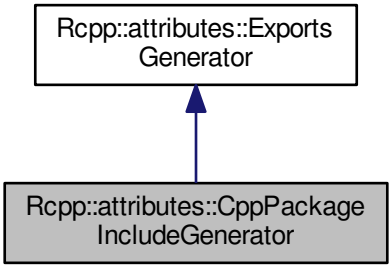
- inst/include/Rcpp/Module.h

6.118 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 &fileSep)
- virtual void writeBegin ()
- virtual void writeEnd ()
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)
- std::string getHeaderGuard () const

Private Attributes

- std::string includeDir_

Additional Inherited Members

6.118.1 Detailed Description

Definition at line 688 of file attributes.cpp.

6.118.2 Constructor & Destructor Documentation

6.118.2.1 Rcpp::attributes::CppPackageIncludeGenerator::CppPackageIncludeGenerator (const std::string & *packageDir,* const std::string & *package,* const std::string & *fileSep*)

Definition at line 2155 of file attributes.cpp.

References includeDir_.

6.118.3 Member Function Documentation

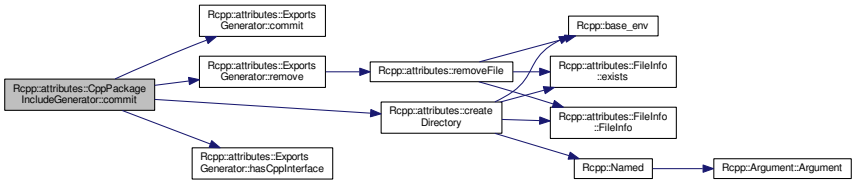
6.118.3.1 bool Rcpp::attributes::CppPackageIncludeGenerator::commit (const std::vector< std::string > & *includes*) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2183 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:



6.118.3.2 virtual void Rcpp::attributes::CppPackageIncludeGenerator::doWriteFunctions (const SourceFileAttributes & *attributes,* bool *verbose*) [inline], [private], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 699 of file attributes.cpp.

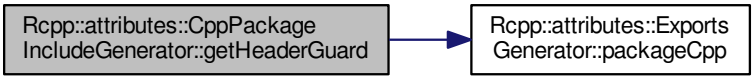
6.118.3.3 std::string Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard ( ) const [private]

Definition at line 2199 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::packageCpp().

Referenced by writeEnd().

Here is the call graph for this function:



6.118.3.4 virtual void Rcpp::attributes::CppPackageIncludeGenerator::writeBegin ( ) [inline], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 694 of file attributes.cpp.

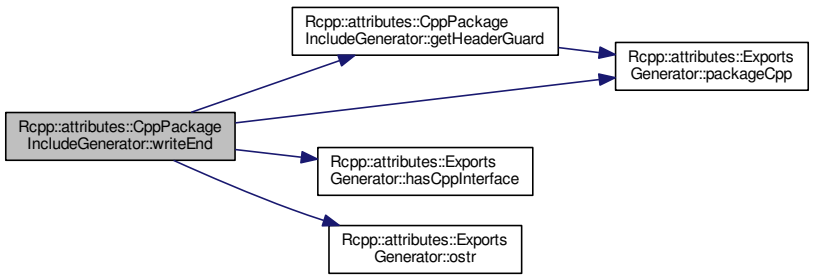
6.118.3.5 void Rcpp::attributes::CppPackageIncludeGenerator::writeEnd ( ) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2168 of file attributes.cpp.

References getHeaderGuard(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), Rcpp::attributes::Exports↩ Generator::ostr(), and Rcpp::attributes::ExportsGenerator::packageCpp().

Here is the call graph for this function:



6.118.4 Member Data Documentation

6.118.4.1 std::string Rcpp::attributes::CppPackageIncludeGenerator::includeDir_ [private]

Definition at line 704 of file attributes.cpp.

Referenced by commit(), and CppPackageIncludeGenerator().

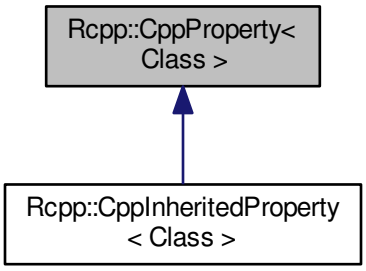
The documentation for this class was generated from the following file:

- src/attributes.cpp

6.119 Rcpp::CppProperty< Class > Class Template Reference

`#include <Module.h>`

Inheritance diagram for Rcpp::CppProperty< Class >:



Public Types

- typedef Rcpp::XPtr< Class > XP

Public Member Functions

- CppProperty (const char *doc=0)
- virtual ~CppProperty ()
- virtual SEXP get (Class *)
- virtual void set (Class *, SEXP)
- virtual bool is_readonly ()
- virtual std::string get_class ()

Public Attributes

- std::string docstring

6.119.1 Detailed Description

template<typename Class>
class Rcpp::CppProperty< Class >

Definition at line 266 of file Module.h.

6.119.2 Member Typedef Documentation

6.119.2.1 template<typename Class> typedef Rcpp::XPtr<Class> Rcpp::CppProperty< Class >::XP

Definition at line 268 of file Module.h.

6.119.3 Constructor & Destructor Documentation

6.119.3.1 template<typename Class> Rcpp::CppProperty< Class >::CppProperty (const char * *doc* = 0) [inline]

Definition at line 270 of file Module.h.

6.119.3.2 template<typename Class> virtual Rcpp::CppProperty< Class >::~CppProperty ( ) [inline], [virtual]

Definition at line 271 of file Module.h.

6.119.4 Member Function Documentation

6.119.4.1 template<typename Class> virtual SEXP Rcpp::CppProperty< Class >::get (Class *) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedProperty< Class, Parent >.

Definition at line 272 of file Module.h.

6.119.4.2 template<typename Class> virtual std::string Rcpp::CppProperty< Class >::get_class ( ) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedProperty< Class, Parent >.

Definition at line 275 of file Module.h.

Referenced by Rcpp::S4_field< Class >::S4_field().

6.119.4.3 template<typename Class> virtual bool Rcpp::CppProperty< Class >::is_readonly ( ) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedProperty< Class, Parent >.

Definition at line 274 of file Module.h.

Referenced by Rcpp::S4_field< Class >::S4_field().

6.119.4.4 template<typename Class> virtual void Rcpp::CppProperty< Class >::set (Class *, SEXP) [inline], [virtual]

Reimplemented in Rcpp::CppInheritedProperty< Class, Parent >.

Definition at line 273 of file Module.h.

6.119.5 Member Data Documentation

6.119.5.1 template<typename Class> std::string Rcpp::CppProperty< Class >::docstring

Definition at line 277 of file Module.h.

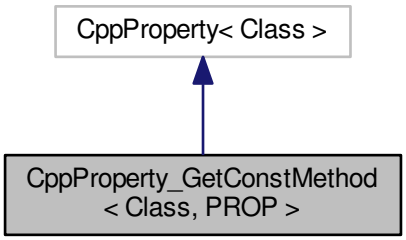
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

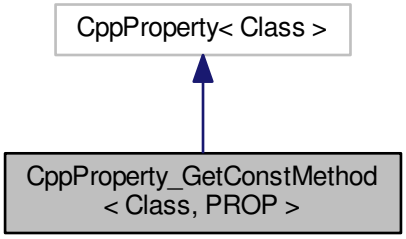
6.120 CppProperty_GetConstMethod< Class, PROP > Class Template Reference

```
#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

6.120.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod< Class, PROP >

Definition at line 48 of file Module_Property.h.

6.120.2 Member Typedef Documentation

6.120.2.1 template<typename Class , typename PROP > typedef PROP(Class::* CppProperty_GetConstMethod< Class, PROP >::GetMethod) (void) const

Definition at line 50 of file Module_Property.h.

6.120.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetConstMethod< Class, PROP >::prop_class

Definition at line 51 of file Module_Property.h.

6.120.3 Constructor & Destructor Documentation

6.120.3.1 template<typename Class , typename PROP > CppProperty_GetConstMethod< Class, PROP >::CppProperty_GetConstMethod (GetMethod *getter_*, const char * *doc* = 0) [inline]

Definition at line 53 of file Module_Property.h.

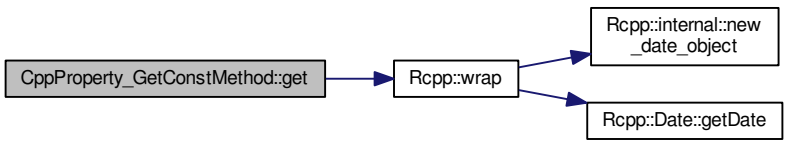
6.120.4 Member Function Documentation

6.120.4.1 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_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.120.4.2 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_GetMethod< Class, PROP >::class_name.

6.120.4.3 template<typename Class , typename PROP > bool CppProperty_GetConstMethod< Class, PROP >::is_readonly ( ) [inline]

Definition at line 58 of file Module_Property.h.

6.120.4.4 template<typename Class , typename PROP > void CppProperty_GetConstMethod< Class, PROP >::set (Class *, SEXP) [inline]

Definition at line 57 of file Module_Property.h.

6.120.5 Member Data Documentation

6.120.5.1 template<typename Class , typename PROP > std::string CppProperty_GetConstMethod< Class, PROP >::class_name [private]

Definition at line 63 of file Module_Property.h.

6.120.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetConstMethod< Class, PROP >::getter [private]

Definition at line 62 of file Module_Property.h.

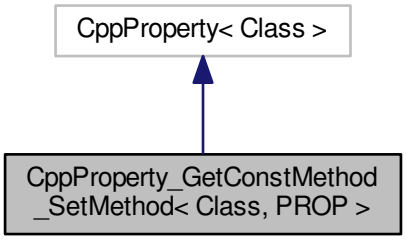
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

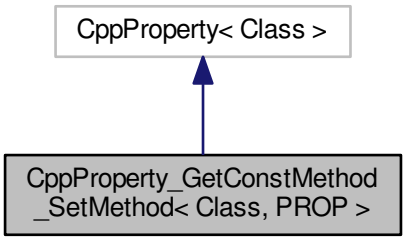
6.121 CppProperty_GetConstMethod_SetMethod< Class, PROP > Class Template Reference

`#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

6.121.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetMethod< Class, PROP >

Definition at line 117 of file Module_Property.h.

6.121.2 Member Typedef Documentation

6.121.2.1 template<typename Class , typename PROP > typedef PROP(Class::∗ CppProperty_GetConstMethod_Set↩ Method< Class, PROP >::GetMethod) (void) const

Definition at line 119 of file Module_Property.h.

6.121.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetConstMethod_↩ SetMethod< Class, PROP >::prop_class

Definition at line 121 of file Module_Property.h.

6.121.2.3 template<typename Class , typename PROP > typedef void(Class::∗ CppProperty_GetConstMethod_Set↩ Method< Class, PROP >::SetMethod) (PROP)

Definition at line 120 of file Module_Property.h.

6.121.3 Constructor & Destructor Documentation

6.121.3.1 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 123 of file Module_Property.h.

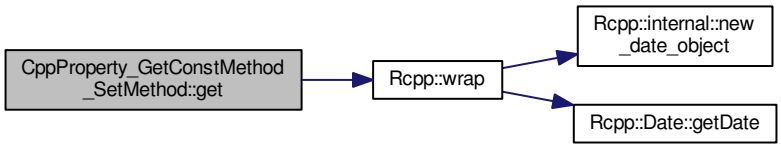
6.121.4 Member Function Documentation

6.121.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetConstMethod_SetMethod< Class, PROP >::get (Class * *object*) [inline]

Definition at line 126 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.121.4.2 template<typename Class , typename PROP > std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::get_class ( ) [inline]

Definition at line 135 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.121.4.3 template<typename Class , typename PROP > bool CppProperty_GetConstMethod_SetMethod< Class, PROP >::is_readonly ( ) [inline]

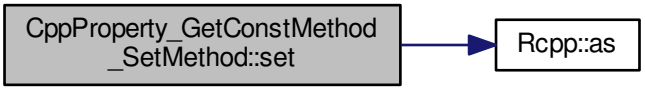
Definition at line 134 of file Module_Property.h.

6.121.4.4 template<typename Class , typename PROP > void CppProperty_GetConstMethod_SetMethod< Class, PROP >::set (Class * *object,* SEXP *value*) [inline]

Definition at line 129 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.121.5 Member Data Documentation

6.121.5.1 template<typename Class , typename PROP > std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::class_name [private]

Definition at line 140 of file Module_Property.h.

6.121.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::getter [private]

Definition at line 138 of file Module_Property.h.

6.121.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::setter [private]

Definition at line 139 of file Module_Property.h.

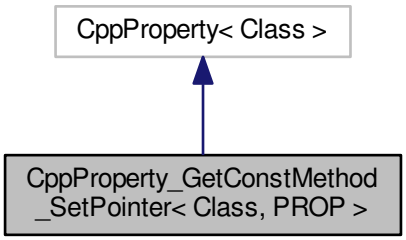
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

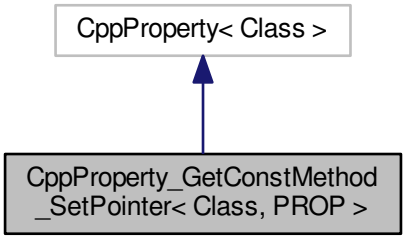
6.122 CppProperty_GetConstMethod_SetPointer< Class, PROP > Class Template Reference

`#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

6.122.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetPointer< Class, PROP >

Definition at line 176 of file Module_Property.h.

6.122.2 Member Typedef Documentation

6.122.2.1 template<typename Class , typename PROP > typedef PROP(Class::* CppProperty_GetConstMethod_SetPointer< Class, PROP >::GetMethod) (void) const

Definition at line 178 of file Module_Property.h.

6.122.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetConstMethod_SetPointer< Class, PROP >::prop_class

Definition at line 180 of file Module_Property.h.

6.122.2.3 template<typename Class , typename PROP > typedef void(* CppProperty_GetConstMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)

Definition at line 179 of file Module_Property.h.

6.122.3 Constructor & Destructor Documentation

6.122.3.1 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 182 of file Module_Property.h.

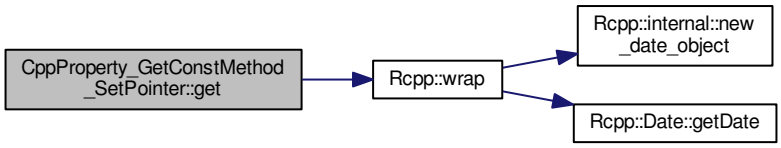
6.122.4 Member Function Documentation

6.122.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetConstMethod_SetPointer< Class, PROP >::get (Class * *object*) [inline]

Definition at line 185 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.122.4.2 template<typename Class , typename PROP > std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::get_class ( ) [inline]

Definition at line 194 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.122.4.3 template<typename Class , typename PROP > bool CppProperty_GetConstMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]

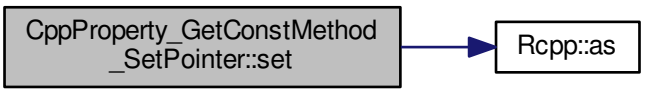
Definition at line 193 of file Module_Property.h.

6.122.4.4 template<typename Class , typename PROP > void CppProperty_GetConstMethod_SetPointer< Class, PROP >::set (Class * *object,* SEXP *value*) [inline]

Definition at line 188 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.122.5 Member Data Documentation

6.122.5.1 template<typename Class , typename PROP > std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::class_name [private]

Definition at line 199 of file Module_Property.h.

6.122.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::getter [private]

Definition at line 197 of file Module_Property.h.

6.122.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::setter [private]

Definition at line 198 of file Module_Property.h.

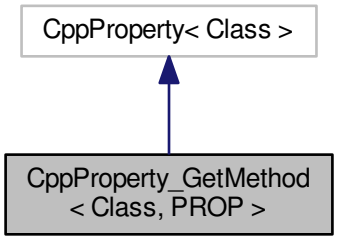
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

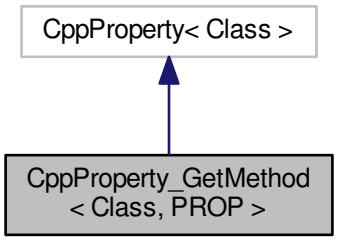
6.123 CppProperty_GetMethod< Class, PROP > Class Template Reference

`#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

6.123.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetMethod< Class, PROP >

Definition at line 27 of file Module_Property.h.

6.123.2 Member Typedef Documentation

6.123.2.1 template<typename Class , typename PROP > typedef PROP(Class::* CppProperty_GetMethod< Class, PROP >::GetMethod) (void)

Definition at line 29 of file Module_Property.h.

6.123.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetMethod< Class, PROP >::prop_class

Definition at line 30 of file Module_Property.h.

6.123.3 Constructor & Destructor Documentation

6.123.3.1 template<typename Class , typename PROP > CppProperty_GetMethod< Class, PROP >::CppProperty_GetMethod (GetMethod *getter_*, const char * *doc* = `0`) [inline]

Definition at line 32 of file Module_Property.h.

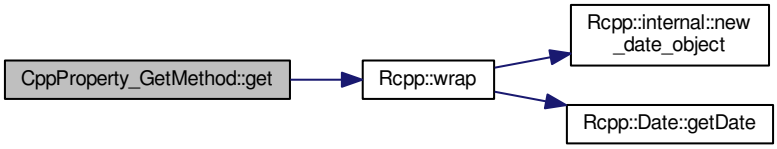
6.123.4 Member Function Documentation

6.123.4.1 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:



6.123.4.2 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.

6.123.4.3 template<typename Class , typename PROP > bool CppProperty_GetMethod< Class, PROP >::is_readonly ( ) [inline]

Definition at line 37 of file Module_Property.h.

6.123.4.4 template<typename Class , typename PROP > void CppProperty_GetMethod< Class, PROP >::set (Class * , SEXP) [inline]

Definition at line 36 of file Module_Property.h.

6.123.5 Member Data Documentation

6.123.5.1 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(), CppProperty_GetConstMethod< Class, PROP >::get_class(), CppProperty_GetPointerMethod< Class, PROP >::get_class(), CppProperty_GetMethod_SetMethod< Class, PROP >::get_class(), CppProperty_GetConstMethod_SetMethod< Class, PROP >::get_class(), CppProperty_GetMethod_SetPointer< Class, PROP >::get_class(), CppProperty_GetConstMethod_SetPointer< Class, PROP >::get_class(), CppProperty_GetPointer_SetMethod< Class, PROP >::get_class(), and CppProperty_GetPointer_SetPointer< Class, PROP >::get_class().

6.123.5.2 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(), 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(), and CppProperty_GetPointer_SetPointer< Class, PROP >::get().

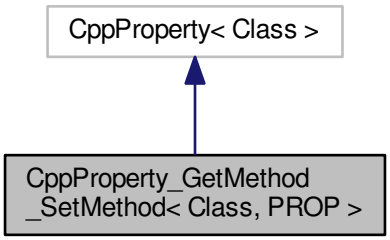
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

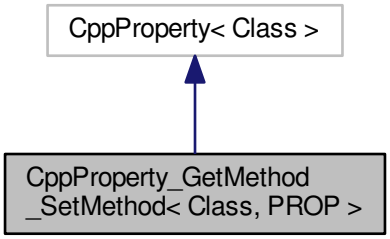
6.124 CppProperty_GetMethod_SetMethod< Class, PROP > Class Template Reference

`#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

- CppProperty_GetMethod_SetMethod (GetMethod getter_, SetMethod setter_, const char *doc=0)
- SEXP get (Class *object)
- void set (Class *object, SEXP value) throw (std::range_error,Rcpp::not_compatible)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.124.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetMethod_SetMethod< Class, PROP >

Definition at line 91 of file Module_Property.h.

6.124.2 Member Typedef Documentation

6.124.2.1 template<typename Class , typename PROP > typedef PROP(Class::* CppProperty_GetMethod_SetMethod< Class, PROP >::GetMethod) (void)

Definition at line 93 of file Module_Property.h.

6.124.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetMethod_Set↩ Method< Class, PROP >::prop_class

Definition at line 95 of file Module_Property.h.

6.124.2.3 template<typename Class , typename PROP > typedef void(Class::* CppProperty_GetMethod_SetMethod< Class, PROP >::SetMethod) (PROP)

Definition at line 94 of file Module_Property.h.

6.124.3 Constructor & Destructor Documentation

6.124.3.1 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.

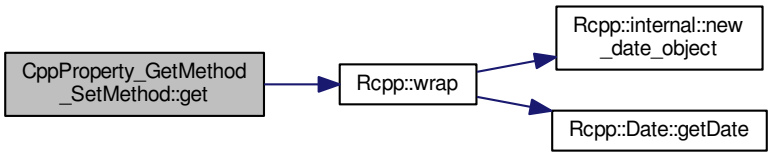
6.124.4 Member Function Documentation

6.124.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetMethod_SetMethod< Class, PROP >::get (Class * *object*) [inline]

Definition at line 100 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.124.4.2 template<typename Class , typename PROP > std::string CppProperty_GetMethod_SetMethod< Class, PROP >::get_class ( ) [inline]

Definition at line 109 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.124.4.3 template<typename Class , typename PROP > bool CppProperty_GetMethod_SetMethod< Class, PROP >::is_readonly ( ) [inline]

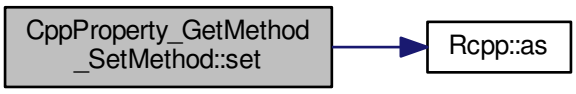
Definition at line 108 of file Module_Property.h.

6.124.4.4 template<typename Class , typename PROP > void CppProperty_GetMethod_SetMethod< Class, PROP >::set (Class * *object,* SEXP *value*) throw std::range_error, Rcpp::not_compatible) [inline]

Definition at line 103 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.124.5 Member Data Documentation

6.124.5.1 template<typename Class , typename PROP > std::string CppProperty_GetMethod_SetMethod< Class, PROP >::class_name [private]

Definition at line 114 of file Module_Property.h.

6.124.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::getter [private]

Definition at line 112 of file Module_Property.h.

6.124.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::setter [private]

Definition at line 113 of file Module_Property.h.

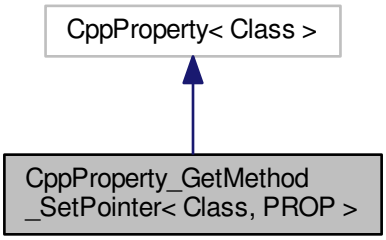
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

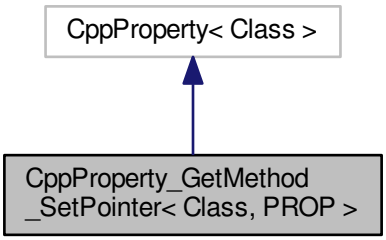
6.125 CppProperty_GetMethod_SetPointer< Class, PROP > Class Template Reference

`#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) throw (std::range_error,Rcpp::not_compatible)
- 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_GetMethod_SetPointer< Class, PROP >

Definition at line 149 of file Module_Property.h.

6.125.2 Member Typedef Documentation

6.125.2.1 template<typename Class , typename PROP > typedef PROP(Class::* CppProperty_GetMethod_SetPointer< Class, PROP >::GetMethod) (void)

Definition at line 151 of file Module_Property.h.

6.125.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetMethod_Set↩ Pointer< Class, PROP >::prop_class

Definition at line 153 of file Module_Property.h.

6.125.2.3 template<typename Class , typename PROP > typedef void(* CppProperty_GetMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)

Definition at line 152 of file Module_Property.h.

6.125.3 Constructor & Destructor Documentation

6.125.3.1 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 155 of file Module_Property.h.

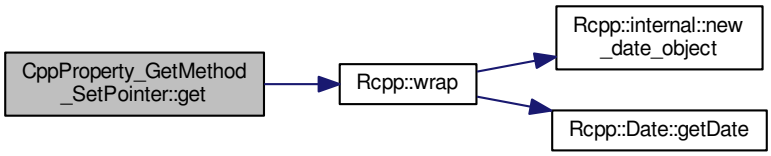
6.125.4 Member Function Documentation

6.125.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetMethod_SetPointer< Class, PROP >::get (Class * *object*) [inline]

Definition at line 158 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.125.4.2 template<typename Class , typename PROP > std::string CppProperty_GetMethod_SetPointer< Class, PROP >::get_class ( ) [inline]

Definition at line 167 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.125.4.3 template<typename Class , typename PROP > bool CppProperty_GetMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]

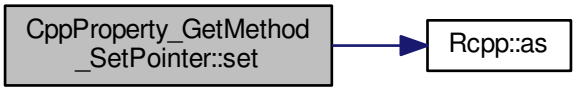
Definition at line 166 of file Module_Property.h.

6.125.4.4 template<typename Class , typename PROP > void CppProperty_GetMethod_SetPointer< Class, PROP >::set (Class * *object*, SEXP *value*) throw std::range_error, Rcpp::not_compatible) [inline]

Definition at line 161 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.125.5 Member Data Documentation

6.125.5.1 template<typename Class , typename PROP > std::string CppProperty_GetMethod_SetPointer< Class, PROP >::class_name [private]

Definition at line 172 of file Module_Property.h.

6.125.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::getter [private]

Definition at line 170 of file Module_Property.h.

6.125.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::setter [private]

Definition at line 171 of file Module_Property.h.

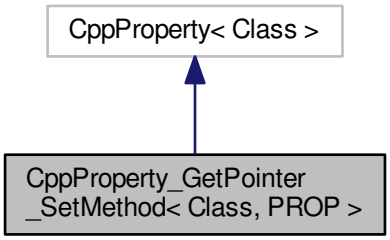
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

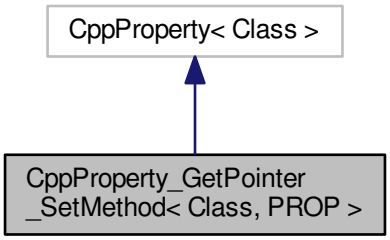
6.126 CppProperty_GetPointer_SetMethod< Class, PROP > Class Template Reference

`#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

6.126.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointer_SetMethod< Class, PROP >

Definition at line 205 of file Module_Property.h.

6.126.2 Member Typedef Documentation

6.126.2.1 template<typename Class , typename PROP > typedef PROP(* CppProperty_GetPointer_SetMethod< Class, PROP >::GetMethod) (Class *)

Definition at line 207 of file Module_Property.h.

6.126.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetPointer_SetMethod< Class, PROP >::prop_class

Definition at line 209 of file Module_Property.h.

6.126.2.3 template<typename Class , typename PROP > typedef void(Class::* CppProperty_GetPointer_SetMethod< Class, PROP >::SetMethod) (PROP)

Definition at line 208 of file Module_Property.h.

6.126.3 Constructor & Destructor Documentation

6.126.3.1 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 211 of file Module_Property.h.

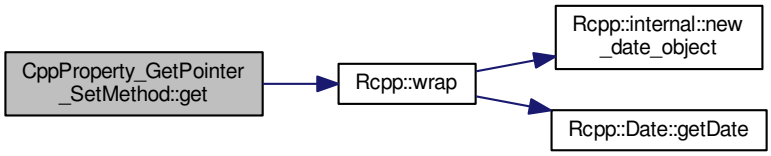
6.126.4 Member Function Documentation

6.126.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetPointer_SetMethod< Class, PROP >::get (Class * *object*) [inline]

Definition at line 214 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.126.4.2 template<typename Class , typename PROP > std::string CppProperty_GetPointer_SetMethod< Class, PROP >::get_class ( ) [inline]

Definition at line 223 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.126.4.3 template<typename Class , typename PROP > bool CppProperty_GetPointer_SetMethod< Class, PROP >::is_readonly ( ) [inline]

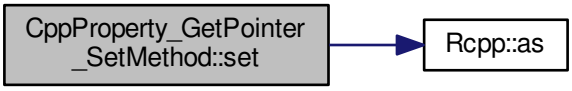
Definition at line 222 of file Module_Property.h.

6.126.4.4 template<typename Class , typename PROP > void CppProperty_GetPointer_SetMethod< Class, PROP >::set (Class * *object,* SEXP *value*) [inline]

Definition at line 217 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.126.5 Member Data Documentation

6.126.5.1 template<typename Class , typename PROP > std::string CppProperty_GetPointer_SetMethod< Class, PROP >::class_name [private]

Definition at line 228 of file Module_Property.h.

6.126.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::getter [private]

Definition at line 226 of file Module_Property.h.

6.126.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::setter [private]

Definition at line 227 of file Module_Property.h.

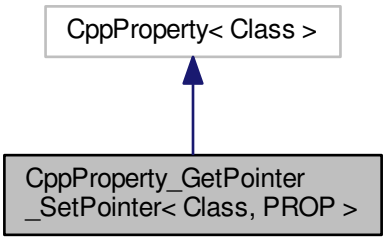
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

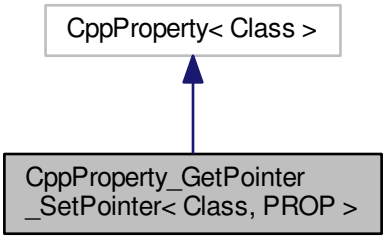
6.127 CppProperty_GetPointer_SetPointer< Class, PROP > Class Template Reference

`#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

6.127.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointer_SetPointer< Class, PROP >

Definition at line 235 of file Module_Property.h.

6.127.2 Member Typedef Documentation

6.127.2.1 template<typename Class , typename PROP > typedef PROP(* CppProperty_GetPointer_SetPointer< Class, PROP >::GetMethod) (Class *)

Definition at line 237 of file Module_Property.h.

6.127.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetPointer_Set↩ Pointer< Class, PROP >::prop_class

Definition at line 239 of file Module_Property.h.

6.127.2.3 template<typename Class , typename PROP > typedef void(* CppProperty_GetPointer_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)

Definition at line 238 of file Module_Property.h.

6.127.3 Constructor & Destructor Documentation

6.127.3.1 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 241 of file Module_Property.h.

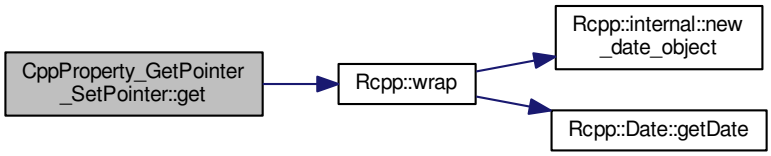
6.127.4 Member Function Documentation

6.127.4.1 template<typename Class , typename PROP > SEXP CppProperty_GetPointer_SetPointer< Class, PROP >::get (Class * *object*) [inline]

Definition at line 244 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 template<typename Class , typename PROP > std::string CppProperty_GetPointer_SetPointer< Class, PROP >::get_class ( ) [inline]

Definition at line 253 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.127.4.3 template<typename Class , typename PROP > bool CppProperty_GetPointer_SetPointer< Class, PROP >::is_readonly ( ) [inline]

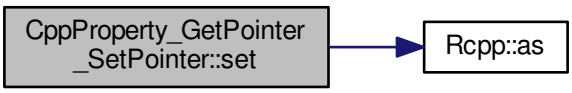
Definition at line 252 of file Module_Property.h.

6.127.4.4 template<typename Class , typename PROP > void CppProperty_GetPointer_SetPointer< Class, PROP >::set (Class * *object,* SEXP *value*) [inline]

Definition at line 247 of file Module_Property.h.

References Rcpp::as().

Here is the call graph for this function:



6.127.5 Member Data Documentation

6.127.5.1 template<typename Class , typename PROP > std::string CppProperty_GetPointer_SetPointer< Class, PROP >::class_name [private]

Definition at line 258 of file Module_Property.h.

6.127.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::getter [private]

Definition at line 256 of file Module_Property.h.

6.127.5.3 template<typename Class , typename PROP > SetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::setter [private]

Definition at line 257 of file Module_Property.h.

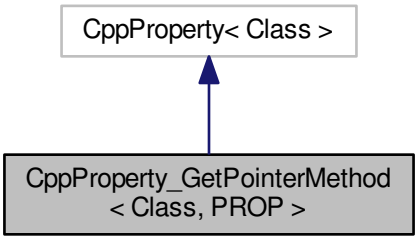
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

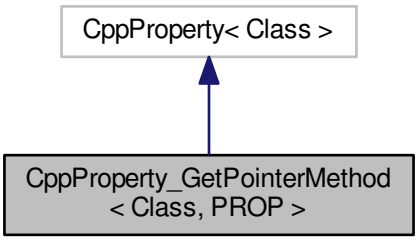
6.128 CppProperty_GetPointerMethod< Class, PROP > Class Template Reference

`#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

6.128.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointerMethod< Class, PROP >

Definition at line 70 of file Module_Property.h.

6.128.2 Member Typedef Documentation

6.128.2.1 template<typename Class , typename PROP > typedef PROP(∗ CppProperty_GetPointerMethod< Class, PROP >::GetMethod) (Class ∗)

Definition at line 72 of file Module_Property.h.

6.128.2.2 template<typename Class , typename PROP > typedef CppProperty<Class> CppProperty_GetPointerMethod< Class, PROP >::prop_class

Definition at line 73 of file Module_Property.h.

6.128.3 Constructor & Destructor Documentation

6.128.3.1 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.

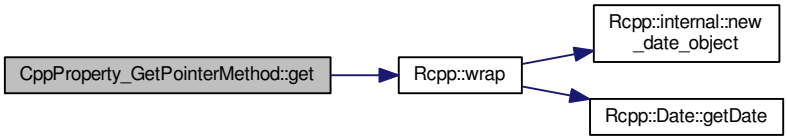
6.128.4 Member Function Documentation

6.128.4.1 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_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.128.4.2 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_GetMethod< Class, PROP >::class_name.

6.128.4.3 template<typename Class , typename PROP > bool CppProperty_GetPointerMethod< Class, PROP >::is_readonly ( ) [inline]

Definition at line 80 of file Module_Property.h.

6.128.4.4 template<typename Class , typename PROP > void CppProperty_GetPointerMethod< Class, PROP >::set (Class *, SEXP) [inline]

Definition at line 79 of file Module_Property.h.

6.128.5 Member Data Documentation

6.128.5.1 template<typename Class , typename PROP > std::string CppProperty_GetPointerMethod< Class, PROP >::class_name [private]

Definition at line 85 of file Module_Property.h.

6.128.5.2 template<typename Class , typename PROP > GetMethod CppProperty_GetPointerMethod< Class, PROP >::getter [private]

Definition at line 84 of file Module_Property.h.

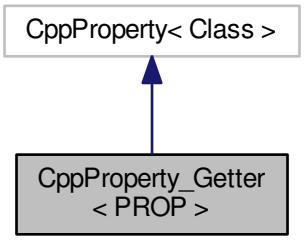
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

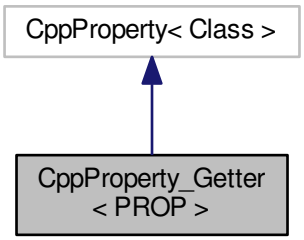
6.129 CppProperty_Getter< PROP > Class Template Reference

`#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

6.129.1 Detailed Description

template<typename PROP>
class CppProperty_Getter< PROP >

Definition at line 48 of file Module_Field.h.

6.129.2 Member Typedef Documentation

6.129.2.1 template<typename PROP > typedef PROP Class::* CppProperty_Getter< PROP >::pointer

Definition at line 50 of file Module_Field.h.

6.129.2.2 template<typename PROP > typedef CppProperty<Class> CppProperty_Getter< PROP >::prop_class

Definition at line 51 of file Module_Field.h.

6.129.3 Constructor & Destructor Documentation

6.129.3.1 template<typename PROP > CppProperty_Getter< PROP >::CppProperty_Getter (pointer *ptr_*, const char * *doc* = `0`) [inline]

Definition at line 53 of file Module_Field.h.

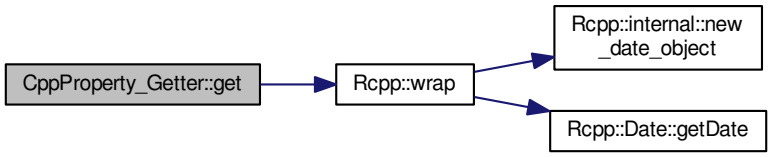
6.129.4 Member Function Documentation

6.129.4.1 template<typename PROP > SEXP CppProperty_Getter< PROP >::get (Class * *object*) [inline]

Definition at line 56 of file Module_Field.h.

References CppProperty_Getter_Setter< PROP >::ptr, and Rcpp::wrap().

Here is the call graph for this function:



6.129.4.2 template<typename PROP > std::string CppProperty_Getter< PROP >::get_class ( ) [inline]

Definition at line 59 of file Module_Field.h.

References CppProperty_Getter_Setter< PROP >::class_name.

6.129.4.3 template<typename PROP > bool CppProperty_Getter< PROP >::is_readonly ( ) [inline]

Definition at line 58 of file Module_Field.h.

6.129.4.4 template<typename PROP > void CppProperty_Getter< PROP >::set (Class * *object*, SEXP *value*) [inline]

Definition at line 57 of file Module_Field.h.

6.129.5 Member Data Documentation

6.129.5.1 template<typename PROP > std::string CppProperty_Getter< PROP >::class_name [private]

Definition at line 63 of file Module_Field.h.

6.129.5.2 template<typename PROP > pointer CppProperty_Getter< PROP >::ptr [private]

Definition at line 62 of file Module_Field.h.

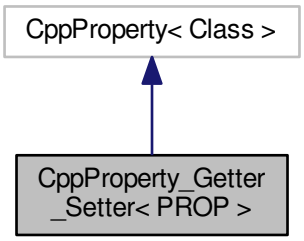
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Field.h

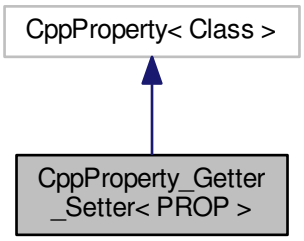
6.130 CppProperty_Getter_Setter< PROP > Class Template Reference

`#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

6.130.1 Detailed Description

template<typename PROP>
class CppProperty_Getter_Setter< PROP >

Definition at line 27 of file Module_Field.h.

6.130.2 Member Typedef Documentation

6.130.2.1 template<typename PROP > typedef PROP Class::* CppProperty_Getter_Setter< PROP >::pointer

Definition at line 29 of file Module_Field.h.

6.130.2.2 template<typename PROP > typedef CppProperty<Class> CppProperty_Getter_Setter< PROP >::prop_class

Definition at line 30 of file Module_Field.h.

6.130.3 Constructor & Destructor Documentation

6.130.3.1 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.

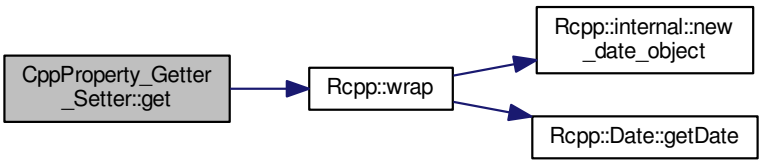
6.130.4 Member Function Documentation

6.130.4.1 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:



6.130.4.2 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.

6.130.4.3 template<typename PROP > bool CppProperty_Getter_Setter< PROP >::is_readonly ( ) [inline]

Definition at line 37 of file Module_Field.h.

6.130.4.4 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.

6.130.5 Member Data Documentation

6.130.5.1 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(), and CppProperty_Getter< PROP >::get_class().

6.130.5.2 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(), CppProperty_Getter< 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

6.131 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 ()

6.131.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::ctype< T >

Definition at line 82 of file algorithm.h.

6.131.2 Member Typedef Documentation

6.131.2.1 template<typename T > typedef ctype_helper< sizeof(test(make())) >::type Rcpp::algorithm::helpers::ctype< T >::type

Definition at line 106 of file algorithm.h.

6.131.3 Member Function Documentation

6.131.3.1 template<typename T > static T Rcpp::algorithm::helpers::ctype< T >::make ( ) [static]

6.131.3.2 template<typename T > static CTYPE_CHAR Rcpp::algorithm::helpers::ctype< T >::test (const char &) [static]

6.131.3.3 template<typename T > static CTYPE_SHORT Rcpp::algorithm::helpers::ctype< T >::test (const short &) [static]

6.131.3.4 template<typename T > static CTYPE_INT Rcpp::algorithm::helpers::ctype< T >::test (const int &) [static]

6.131.3.5 template<typename T > static CTYPE_LONG Rcpp::algorithm::helpers::ctype< T >::test (const long &) [static]

6.131.3.6 template<typename T > static CTYPE_FLOAT Rcpp::algorithm::helpers::ctype< T >::test (const float &) [static]

6.131.3.7 template<typename T > static CTYPE_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (const double &) [static]

6.131.3.8 template<typename T > static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (const long double &) [static]

6.131.3.9 template<typename T > static CTYPE_STRING Rcpp::algorithm::helpers::ctype< T >::test (const std::string &) [static]

6.131.3.10 template<typename T > static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::ctype< T >::test (const unsigned char &) [static]

6.131.3.11 template<typename T > static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::ctype< T >::test (const unsigned short &) [static]

6.131.3.12 template<typename T > static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::ctype< T >::test (const unsigned int &) [static]

6.131.3.13 template<typename T > static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::ctype< T >::test (const unsigned long &) [static]

6.131.3.14 template<typename T > static CTYPE_UNKNOWN Rcpp::algorithm::helpers::ctype< T >::test (...) [static]

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.132 Rcpp::algorithm::helpers::CTYPE_CHAR Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [1]

6.132.1 Detailed Description

Definition at line 11 of file algorithm.h.

6.132.2 Member Data Documentation

6.132.2.1 char Rcpp::algorithm::helpers::CTYPE_CHAR::a[1]

Definition at line 11 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.133 Rcpp::algorithm::helpers::CTYPE_DOUBLE Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [7]

6.133.1 Detailed Description

Definition at line 19 of file algorithm.h.

6.133.2 Member Data Documentation

6.133.2.1 char Rcpp::algorithm::helpers::CTYPE_DOUBLE::a[7]

Definition at line 19 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.134 Rcpp::algorithm::helpers::CTYPE_FLOAT Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [6]

6.134.1 Detailed Description

Definition at line 18 of file algorithm.h.

6.134.2 Member Data Documentation

6.134.2.1 char Rcpp::algorithm::helpers::CTYPE_FLOAT::a[6]

Definition at line 18 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.135 Rcpp::algorithm::helpers::ctype_helper< I > Struct Template Reference

`#include <algorithm.h>`

Static Public Attributes

- static const bool value = false

6.135.1 Detailed Description

template<std::size_t I>
struct Rcpp::algorithm::helpers::ctype_helper< I >

Definition at line 32 of file algorithm.h.

6.135.2 Member Data Documentation

6.135.2.1 template<std::size_t I> const bool Rcpp::algorithm::helpers::ctype_helper< I >::value = false [static]

Definition at line 32 of file algorithm.h.

Referenced by Rcpp::algorithm::max(), Rcpp::algorithm::max_nona(), Rcpp::algorithm::mean(), Rcpp::algorithm::min(), Rcpp::algorithm::min_nona(), Rcpp::algorithm::helpers::sqrt::operator()(), Rcpp::algorithm::prod(), Rcpp::algorithm::prod_nona(), Rcpp::algorithm::sum(), and Rcpp::algorithm::sum_nona().

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.136 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef char type

Static Public Attributes

- static const bool value = true

6.136.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >

Definition at line 35 of file algorithm.h.

6.136.2 Member Typedef Documentation

6.136.2.1 typedef char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::type

Definition at line 35 of file algorithm.h.

6.136.3 Member Data Documentation

6.136.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::value = true [static]

Definition at line 35 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_helper< sizeof(CTYPE_DOUBLE) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef double type

Static Public Attributes

- static const bool value = true

6.137.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >

Definition at line 55 of file algorithm.h.

6.137.2 Member Typedef Documentation

6.137.2.1 typedef double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::type

Definition at line 55 of file algorithm.h.

6.137.3 Member Data Documentation

6.137.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::value = true [static]

Definition at line 55 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.138 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef float type

Static Public Attributes

- static const bool value = true

6.138.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >

Definition at line 52 of file algorithm.h.

6.138.2 Member Typedef Documentation

6.138.2.1 typedef float Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::type

Definition at line 52 of file algorithm.h.

6.138.3 Member Data Documentation

6.138.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::value = true [static]

Definition at line 52 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< sizeof(CTYPE_INT) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef int type

Static Public Attributes

- static const bool value = true

6.139.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >

Definition at line 41 of file algorithm.h.

6.139.2 Member Typedef Documentation

6.139.2.1 typedef int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::type

Definition at line 41 of file algorithm.h.

6.139.3 Member Data Documentation

6.139.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::value = true [static]

Definition at line 41 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_LONG) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef long type

Static Public Attributes

- static const bool value = true

6.140.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >

Definition at line 44 of file algorithm.h.

6.140.2 Member Typedef Documentation

6.140.2.1 typedef long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::type

Definition at line 44 of file algorithm.h.

6.140.3 Member Data Documentation

6.140.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::value = true [static]

Definition at line 44 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.141 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef long double type

Static Public Attributes

- static const bool value = true

6.141.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >

Definition at line 58 of file algorithm.h.

6.141.2 Member Typedef Documentation

6.141.2.1 typedef long double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::type

Definition at line 58 of file algorithm.h.

6.141.3 Member Data Documentation

6.141.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::value = true [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

6.142 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) > Struct Template Reference

```
#include <algorithm.h>
```

Public Types

- typedef short type

Static Public Attributes

- static const bool value = true

6.142.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >

Definition at line 38 of file algorithm.h.

6.142.2 Member Typedef Documentation

6.142.2.1 typedef short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::type

Definition at line 38 of file algorithm.h.

6.142.3 Member Data Documentation

6.142.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::value = true [static]

Definition at line 38 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.143 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef std::string type

Static Public Attributes

- static const bool value = true

6.143.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >

Definition at line 61 of file algorithm.h.

6.143.2 Member Typedef Documentation

6.143.2.1 typedef std::string Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::type

Definition at line 61 of file algorithm.h.

6.143.3 Member Data Documentation

6.143.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::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

6.144 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef unsigned char type

Static Public Attributes

- static const bool value = true

6.144.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >

Definition at line 64 of file algorithm.h.

6.144.2 Member Typedef Documentation

6.144.2.1 typedef unsigned char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::type

Definition at line 64 of file algorithm.h.

6.144.3 Member Data Documentation

6.144.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::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

6.145 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef unsigned int type

Static Public Attributes

- static const bool value = true

6.145.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >

Definition at line 70 of file algorithm.h.

6.145.2 Member Typedef Documentation

6.145.2.1 typedef unsigned int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::type

Definition at line 70 of file algorithm.h.

6.145.3 Member Data Documentation

6.145.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::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

6.146 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef unsigned long type

Static Public Attributes

- static const bool value = true

6.146.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >

Definition at line 73 of file algorithm.h.

6.146.2 Member Typedef Documentation

6.146.2.1 typedef unsigned long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::type

Definition at line 73 of file algorithm.h.

6.146.3 Member Data Documentation

6.146.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::value = true [static]

Definition at line 73 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_UNSIGNED_SHORT) > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef unsigned short type

Static Public Attributes

- static const bool value = true

6.147.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >

Definition at line 67 of file algorithm.h.

6.147.2 Member Typedef Documentation

6.147.2.1 typedef unsigned short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >::type

Definition at line 67 of file algorithm.h.

6.147.3 Member Data Documentation

6.147.3.1 const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >::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

6.148 Rcpp::algorithm::helpers::CTYPE_INT Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [3]

6.148.1 Detailed Description

Definition at line 13 of file algorithm.h.

6.148.2 Member Data Documentation

6.148.2.1 char Rcpp::algorithm::helpers::CTYPE_INT::a[3]

Definition at line 13 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_LONG Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [4]

6.149.1 Detailed Description

Definition at line 14 of file algorithm.h.

6.149.2 Member Data Documentation

6.149.2.1 char Rcpp::algorithm::helpers::CTYPE_LONG::a[4]

Definition at line 14 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_LONG_DOUBLE Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [8]

6.150.1 Detailed Description

Definition at line 20 of file algorithm.h.

6.150.2 Member Data Documentation

6.150.2.1 char Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE::a[8]

Definition at line 20 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_SHORT Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [2]

6.151.1 Detailed Description

Definition at line 12 of file algorithm.h.

6.151.2 Member Data Documentation

6.151.2.1 char Rcpp::algorithm::helpers::CTYPE_SHORT::a[2]

Definition at line 12 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_STRING Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [9]

6.152.1 Detailed Description

Definition at line 21 of file algorithm.h.

6.152.2 Member Data Documentation

6.152.2.1 char Rcpp::algorithm::helpers::CTYPE_STRING::a[9]

Definition at line 21 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_UNKNOWN Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [128]

6.153.1 Detailed Description

Definition at line 29 of file algorithm.h.

6.153.2 Member Data Documentation

6.153.2.1 char Rcpp::algorithm::helpers::CTYPE_UNKNOWN::a[128]

Definition at line 29 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_UNSIGNED_CHAR Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [10]

6.154.1 Detailed Description

Definition at line 22 of file algorithm.h.

6.154.2 Member Data Documentation

6.154.2.1 char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR::a[10]

Definition at line 22 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_UNSIGNED_INT Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [12]

6.155.1 Detailed Description

Definition at line 24 of file algorithm.h.

6.155.2 Member Data Documentation

6.155.2.1 char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT::a[12]

Definition at line 24 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_UNSIGNED_LONG Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [13]

6.156.1 Detailed Description

Definition at line 25 of file algorithm.h.

6.156.2 Member Data Documentation

6.156.2.1 char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG::a[13]

Definition at line 25 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_UNSIGNED_SHORT Struct Reference

`#include <algorithm.h>`

Public Attributes

- char a [11]

6.157.1 Detailed Description

Definition at line 23 of file algorithm.h.

6.157.2 Member Data Documentation

6.157.2.1 char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT::a[11]

Definition at line 23 of file algorithm.h.

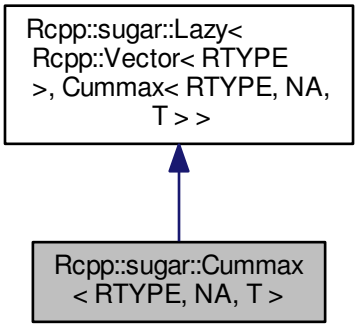
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

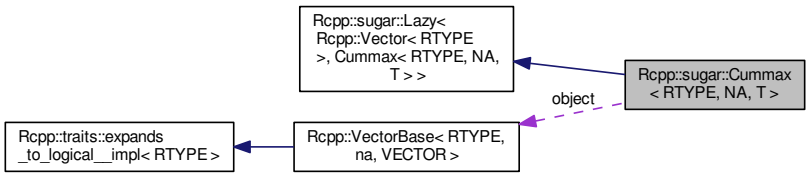
6.158 Rcpp::sugar::Cummax< RTYPE, NA, T > Class Template Reference

`#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

6.158.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cummax< RTYPE, NA, T >

Definition at line 29 of file cummax.h.

6.158.2 Member Typedef Documentation

6.158.2.1 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.

6.158.2.2 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.

6.158.2.3 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.

6.158.3 Constructor & Destructor Documentation

6.158.3.1 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.

6.158.4 Member Function Documentation

6.158.4.1 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.

6.158.5 Member Data Documentation

6.158.5.1 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.

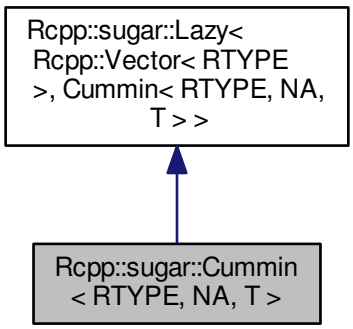
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cummax.h

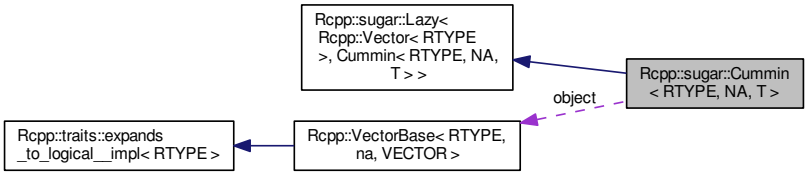
6.159 Rcpp::sugar::Cummin< RTYPE, NA, T > Class Template Reference

`#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

6.159.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cummin< RTYPE, NA, T >

Definition at line 29 of file cummin.h.

6.159.2 Member Typedef Documentation

6.159.2.1 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.

6.159.2.2 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.

6.159.2.3 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.

6.159.3 Constructor & Destructor Documentation

6.159.3.1 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.

6.159.4 Member Function Documentation

6.159.4.1 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.

6.159.5 Member Data Documentation

6.159.5.1 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.

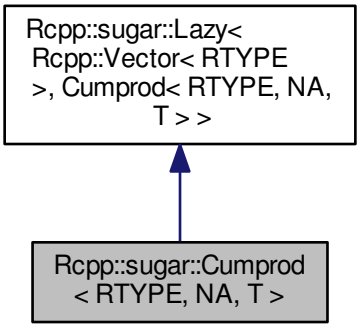
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cummin.h

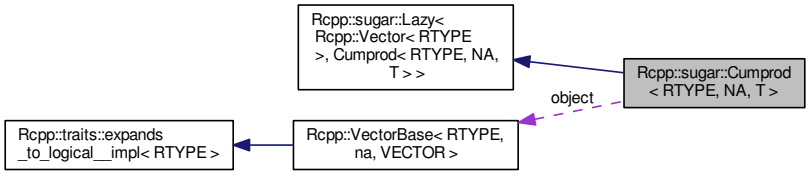
6.160 Rcpp::sugar::Cumprod< RTYPE, NA, T > Class Template Reference

`#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

6.160.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cumprod< RTYPE, NA, T >

Definition at line 29 of file cumprod.h.

6.160.2 Member Typedef Documentation

6.160.2.1 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.

6.160.2.2 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.

6.160.2.3 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.

6.160.3 Constructor & Destructor Documentation

6.160.3.1 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.

6.160.4 Member Function Documentation

6.160.4.1 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.

6.160.5 Member Data Documentation

6.160.5.1 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.

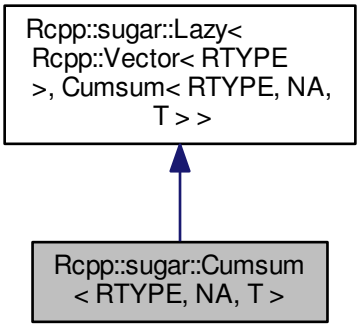
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cumprod.h

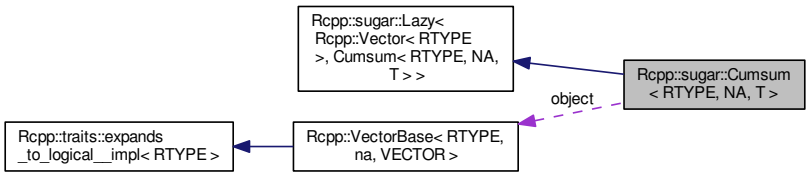
6.161 Rcpp::sugar::Cumsum< RTYPE, NA, T > Class Template Reference

`#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

6.161.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cumsum< RTYPE, NA, T >

Definition at line 29 of file cumsum.h.

6.161.2 Member Typedef Documentation

6.161.2.1 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.

6.161.2.2 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.

6.161.2.3 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.

6.161.3 Constructor & Destructor Documentation

6.161.3.1 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.

6.161.4 Member Function Documentation

6.161.4.1 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.

6.161.5 Member Data Documentation

6.161.5.1 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.

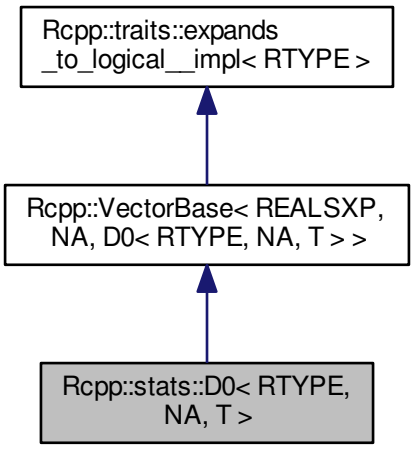
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cumsum.h

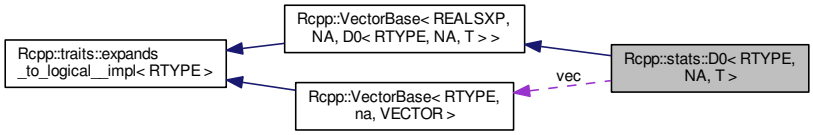
6.162 Rcpp::stats::D0< RTYPE, NA, T > Class Template Reference

`#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

6.162.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D0< RTYPE, NA, T >

Definition at line 33 of file dpq.h.

6.162.2 Member Typedef Documentation

6.162.2.1 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.

6.162.2.2 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.

6.162.3 Constructor & Destructor Documentation

6.162.3.1 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.

6.162.4 Member Function Documentation

6.162.4.1 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.

6.162.4.2 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:

6.162.5 Member Data Documentation

6.162.5.1 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[](), Rcpp::stats::D1< RTYPE, NA, T >::operator[](), Rcpp::stats::D2< RTYPE, NA, T >::operator[](), Rcpp::stats::D3< RTYPE, NA, T >::operator[](), Rcpp::stats::P0< RTYPE, NA, T >::operator[](), Rcpp::stats::P1< RTYPE, NA, T >::operator[](), Rcpp::stats::P2< RTYPE, NA, T >::operator[](), Rcpp::stats::P3< RTYPE, NA, T >::operator[](), Rcpp::stats::Q0< RTYPE, NA, T >::operator[](), Rcpp::stats::Q1< RTYPE, NA, T >::operator[](), Rcpp::stats::Q2< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

6.162.5.2 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[](), Rcpp::stats::D1< RTYPE, NA, T >::operator[](), Rcpp::stats::D2< RTYPE, NA, T >::operator[](), Rcpp::stats::D3< RTYPE, NA, T >::operator[](), Rcpp::stats::P0< RTYPE, NA, T >::operator[](), Rcpp::stats::P1< RTYPE, NA, T >::operator[](), Rcpp::stats::P2< RTYPE, NA, T >::operator[](), Rcpp::stats::P3< RTYPE, NA, T >::operator[](), Rcpp::stats::Q0< RTYPE, NA, T >::operator[](), Rcpp::stats::Q1< RTYPE, NA, T >::operator[](), Rcpp::stats::Q2< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

6.162.5.3 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[](), Rcpp::stats::D1< RTYPE, NA, T >::operator[](), Rcpp::stats::D2< RTYPE, NA, T >::operator[](), Rcpp::stats::D3< RTYPE, NA, T >::operator[](), Rcpp::stats::P0< RTYPE, NA, T >::operator[](), Rcpp::stats::P1< RTYPE, NA, T >::operator[](), Rcpp::stats::P2< RTYPE, NA, T >::operator[](), Rcpp::stats::P3< RTYPE, NA, T >::operator[](), Rcpp::stats::Q0< RTYPE, NA, T >::operator[](), Rcpp::stats::Q1< RTYPE, NA, T >::operator[](), Rcpp::stats::Q2< RTYPE, NA, T >::operator[](), Rcpp::stats::Q3< RTYPE, NA, T >::operator[](), 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(), and Rcpp::stats::Q3< RTYPE, NA, T >::size().

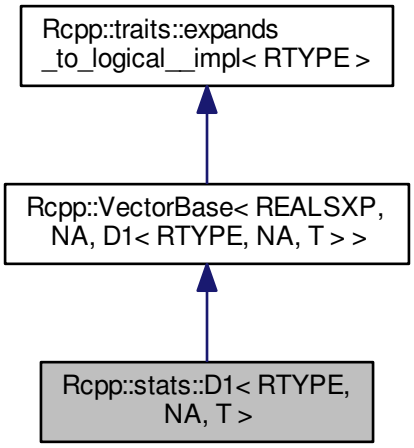
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

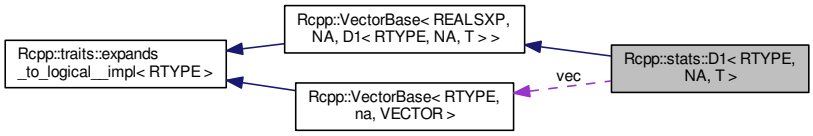
6.163 Rcpp::stats::D1< RTYPE, NA, T > Class Template Reference

`#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

6.163.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D1< RTYPE, NA, T >

Definition at line 54 of file dpq.h.

6.163.2 Member Typedef Documentation

6.163.2.1 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.

6.163.2.2 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.

6.163.3 Constructor & Destructor Documentation

6.163.3.1 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.

6.163.4 Member Function Documentation

6.163.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.163.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.163.5 Member Data Documentation

6.163.5.1 template<int RTYPE, bool NA, typename T> int Rcpp::stats::D1< RTYPE, NA, T >::log [private]

Definition at line 72 of file dpq.h.

6.163.5.2 template<int RTYPE, bool NA, typename T> double Rcpp::stats::D1< RTYPE, NA, T >::p0 [private]

Definition at line 71 of file dpq.h.

6.163.5.3 template<int RTYPE, bool NA, typename T> FunPtr Rcpp::stats::D1< RTYPE, NA, T >::ptr [private]

Definition at line 69 of file dpq.h.

6.163.5.4 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.

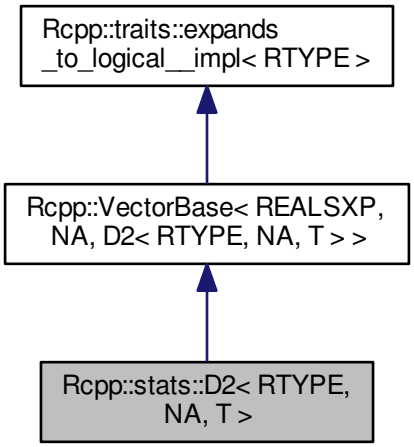
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

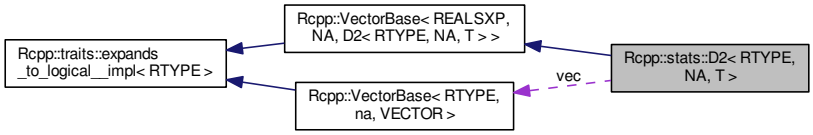
6.164 Rcpp::stats::D2< RTYPE, NA, T > Class Template Reference

`#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

6.164.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D2< RTYPE, NA, T >

Definition at line 76 of file dpq.h.

6.164.2 Member Typedef Documentation

6.164.2.1 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.

6.164.2.2 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.

6.164.3 Constructor & Destructor Documentation

6.164.3.1 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.

6.164.4 Member Function Documentation

6.164.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.164.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.164.5 Member Data Documentation

6.164.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::D2< RTYPE, NA, T >::log [private]

Definition at line 94 of file dpq.h.

6.164.5.2 template<int RTYPE, bool NA, typename T > double Rcpp::stats::D2< RTYPE, NA, T >::p0 [private]

Definition at line 93 of file dpq.h.

6.164.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::D2< RTYPE, NA, T >::p1 [private]

Definition at line 93 of file dpq.h.

6.164.5.4 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::D2< RTYPE, NA, T >::ptr [private]

Definition at line 91 of file dpq.h.

6.164.5.5 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.

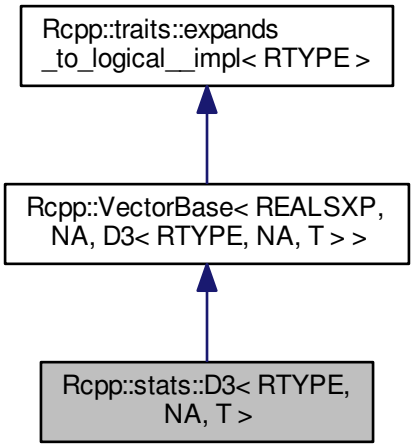
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

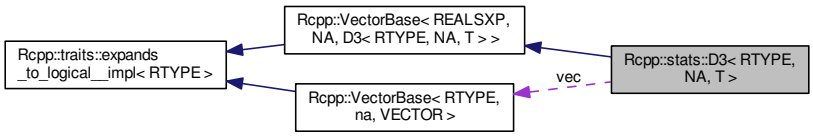
6.165 Rcpp::stats::D3< RTYPE, NA, T > Class Template Reference

`#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

6.165.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D3< RTYPE, NA, T >

Definition at line 98 of file dpq.h.

6.165.2 Member Typedef Documentation

6.165.2.1 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.

6.165.2.2 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.

6.165.3 Constructor & Destructor Documentation

6.165.3.1 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.

6.165.4 Member Function Documentation

6.165.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.165.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.165.5 Member Data Documentation

6.165.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::D3< RTYPE, NA, T >::log [private]

Definition at line 116 of file dpq.h.

6.165.5.2 template<int RTYPE, bool NA, typename T > double Rcpp::stats::D3< RTYPE, NA, T >::p0 [private]

Definition at line 115 of file dpq.h.

6.165.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::D3< RTYPE, NA, T >::p1 [private]

Definition at line 115 of file dpq.h.

6.165.5.4 template<int RTYPE, bool NA, typename T > double Rcpp::stats::D3< RTYPE, NA, T >::p2 [private]

Definition at line 115 of file dpq.h.

6.165.5.5 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::D3< RTYPE, NA, T >::ptr [private]

Definition at line 113 of file dpq.h.

6.165.5.6 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.

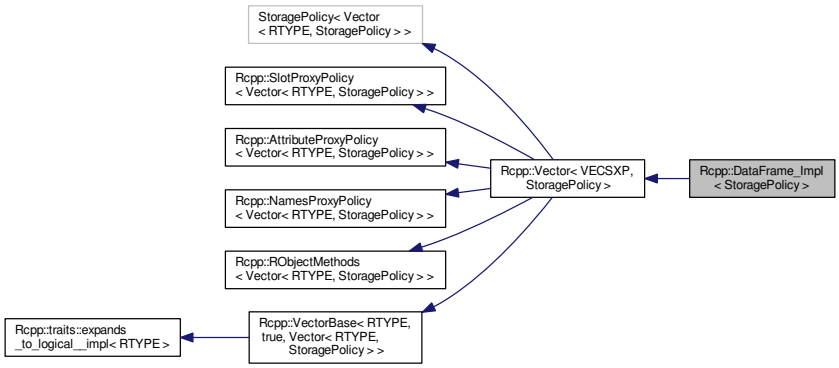
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

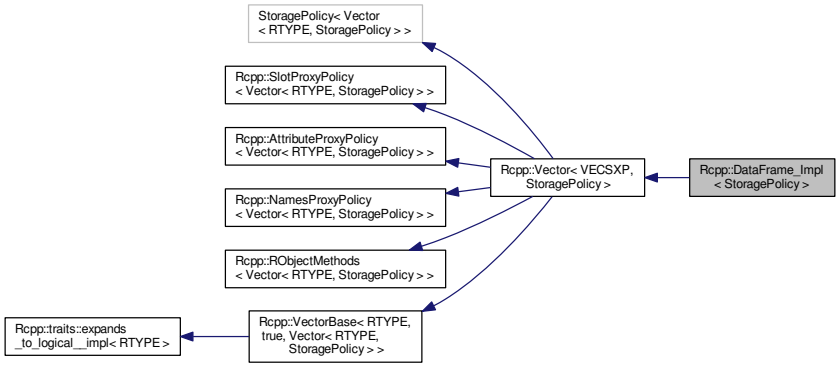
6.166 Rcpp::DataFrame_Impl< StoragePolicy > Class Template Reference

`#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 nrows () const

Static Public Member Functions

- static DataFrame_Impl create ()

Private Member Functions

- void set__ (SEXP x)

Static Private Member Functions

- static DataFrame_Impl from_list (Parent obj)

Additional Inherited Members

6.166.1 Detailed Description

template<template< class > class StoragePolicy>
class Rcpp::DataFrame_Impl< StoragePolicy >

Definition at line 38 of file DataFrame.h.

6.166.2 Member Typedef Documentation

6.166.2.1 template<template< class > class StoragePolicy> typedef Vector<VECSXP, StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::Parent

Definition at line 40 of file DataFrame.h.

6.166.3 Constructor & Destructor Documentation

6.166.3.1 template<template< class > class StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl ( ) [inline]

Definition at line 42 of file DataFrame.h.

6.166.3.2 template<template< class > class StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (SEXP *x*) [inline]

Definition at line 43 of file DataFrame.h.

6.166.3.3 template<template< class > class StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (const DataFrame_Impl< StoragePolicy > & *other*) [inline]

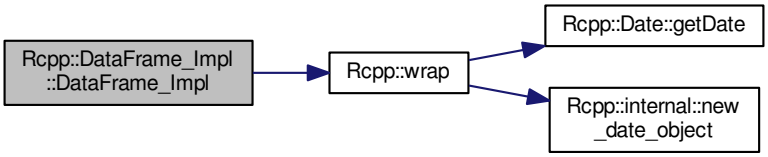
Definition at line 46 of file DataFrame.h.

6.166.3.4 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:



6.166.4 Member Function Documentation

6.166.4.1 template<template< class > class StoragePolicy> static DataFrame_Impl Rcpp::DataFrame_Impl< StoragePolicy >::create ( ) [inline], [static]

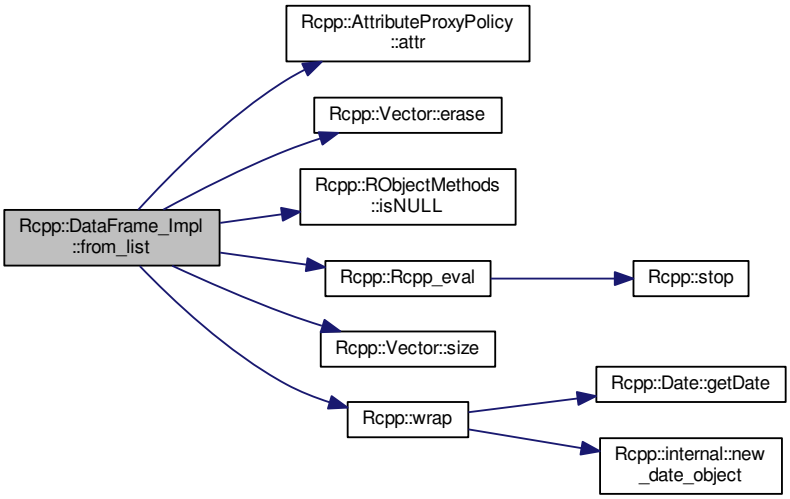
Definition at line 86 of file DataFrame.h.

6.166.4.2 template<template< class > class StoragePolicy> static DataFrame_Impl Rcpp::DataFrame_Impl< StoragePolicy >::from_list (Parent *obj*) [inline], [static], [private]

Definition at line 102 of file DataFrame.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::erase(), Rcpp::R↩ ObjectMethods< Class >::isNULL(), Rcpp::Rcpp_eval(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp↩ ::wrap().

Here is the call graph for this function:



6.166.4.3 template<template< class > class StoragePolicy> int Rcpp::DataFrame_Impl< StoragePolicy >::nrows ( ) const [inline]

Definition at line 69 of file DataFrame.h.

6.166.4.4 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.

6.166.4.5 template<template< class > class StoragePolicy> DataFrame_Impl& Rcpp::DataFrame_Impl< StoragePolicy >::operator= (SEXP *x*) [inline]

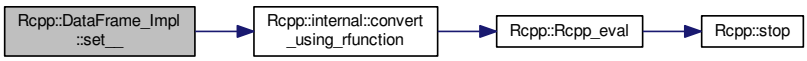
Definition at line 58 of file DataFrame.h.

6.166.4.6 template<template< class > class StoragePolicy> void Rcpp::DataFrame_Impl< StoragePolicy >::set__ (SEXP *x*) [inline], [private]

Definition at line 93 of file DataFrame.h.

References Rcpp::internal::convert_using_rfunction().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/DataFrame.h

6.167 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)
- ~Date ()
- double getDate (void) const
- int getDay () const
- int getMonth () const
- int getYear () const
- int getWeekday () const
- int getYearday () const
- int is_na () 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)

6.167.1 Detailed Description

Definition at line 27 of file Date.h.

6.167.2 Constructor & Destructor Documentation

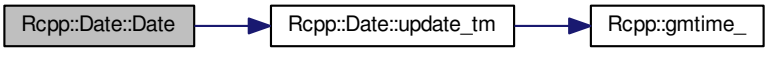
6.167.2.1 Rcpp::Date::Date ( ) [inline]

Definition at line 29 of file Date.h.

References m_d, and update_tm().

Referenced by Date().

Here is the call graph for this function:



6.167.2.2 Rcpp::Date::Date (SEXP *s*) [inline]

Definition at line 25 of file Date.h.

References m_d, and update_tm().

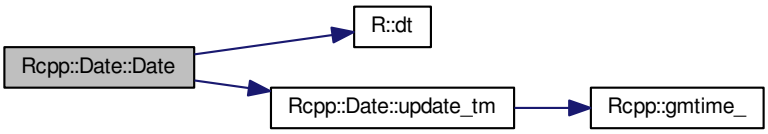
Here is the call graph for this function:

6.167.2.3 Rcpp::Date::Date (const int & *dt*) [inline]

Definition at line 36 of file Date.h.

References R::dt(), m_d, and update_tm().

Here is the call graph for this function:

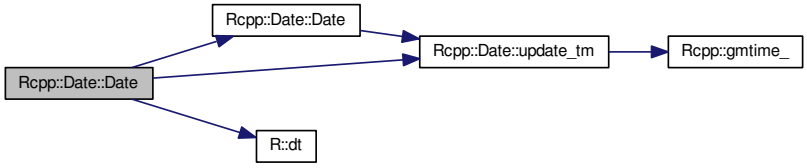


6.167.2.4 Rcpp::Date::Date (const double & *dt*) [inline]

Definition at line 42 of file Date.h.

References Date(), R::dt(), m_d, and update_tm().

Here is the call graph for this function:



Definition at line 30 of file Date.h.

References m_d, and update_tm().

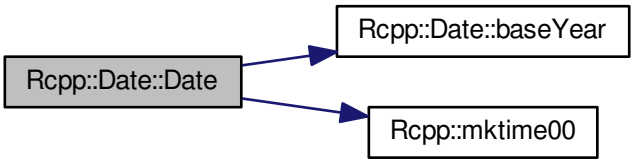
Here is the call graph for this function:

6.167.2.5 Rcpp::Date::Date (const unsigned int & *mon,* const unsigned int & *day,* const unsigned int & *year*) [inline]

Definition at line 48 of file Date.h.

References baseYear(), m_d, m_tm, and Rcpp::mktime00().

Here is the call graph for this function:



6.167.2.6 Rcpp::Date::~Date ( ) [inline]

Definition at line 66 of file Date.h.

6.167.3 Member Function Documentation

6.167.3.1 static unsigned int Rcpp::Date::baseYear ( ) [inline], [static]

Definition at line 83 of file Date.h.

References operator!=, operator+, operator-, operator<, operator<=, operator==, operator>, and operator>=.

Referenced by Date().

6.167.3.2 double Rcpp::Date::getDate (void) const [inline]

Definition at line 68 of file Date.h.

References m_d.

Referenced by Rcpp::wrap().

6.167.3.3 int Rcpp::Date::getDay ( ) const [inline]

Definition at line 76 of file Date.h.

References m_tm.

6.167.3.4 int Rcpp::Date::getMonth ( ) const [inline]

Definition at line 77 of file Date.h.

References m_tm.

6.167.3.5 int Rcpp::Date::getWeekday ( ) const [inline]

Definition at line 79 of file Date.h.

References m_tm.

6.167.3.6 int Rcpp::Date::getYear ( ) const [inline]

Definition at line 78 of file Date.h.

References m_tm.

6.167.3.7 int Rcpp::Date::getYearday ( ) const [inline]

Definition at line 80 of file Date.h.

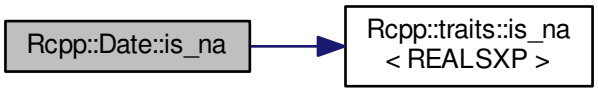
References m_tm.

6.167.3.8 int Rcpp::Date::is_na ( ) const [inline]

Definition at line 97 of file Date.h.

References Rcpp::traits::is_na< REALSXP >(), and m_d.

Here is the call graph for this function:



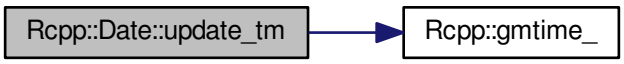
6.167.3.9 void Rcpp::Date::update_tm ( ) [inline], [private]

Definition at line 106 of file Date.h.

References Rcpp::gmtime_(), m_d, and m_tm.

Referenced by Date().

Here is the call graph for this function:



6.167.4 Friends And Related Function Documentation

6.167.4.1 bool operator!= (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 150 of file Date.h.

Referenced by baseYear().

6.167.4.2 Date operator+ (const Date & *date,* int *offset*) [friend]

Definition at line 136 of file Date.h.

Referenced by baseYear().

6.167.4.3 double operator- (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 144 of file Date.h.

Referenced by baseYear().

6.167.4.4 bool operator< (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 145 of file Date.h.

Referenced by baseYear().

6.167.4.5 bool operator<= (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 149 of file Date.h.

Referenced by baseYear().

6.167.4.6 bool operator== (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 147 of file Date.h.

Referenced by baseYear().

6.167.4.7 bool operator> (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 146 of file Date.h.

Referenced by baseYear().

6.167.4.8 bool operator>= (const Date & *date1,* const Date & *date2*) [friend]

Definition at line 148 of file Date.h.

Referenced by baseYear().

6.167.5 Member Data Documentation

6.167.5.1 double Rcpp::Date::m_d [private]

Definition at line 102 of file Date.h.

Referenced by Date(), getDate(), is_na(), Rcpp::operator!=(), Rcpp::operator+(), Rcpp::operator-(), Rcpp::operator<(), Rcpp::operator<=(), Rcpp::operator==(), Rcpp::operator>(), Rcpp::operator>=(), and update_tm().

6.167.5.2 struct tm Rcpp::Date::m_tm [private]

Definition at line 103 of file Date.h.

Referenced by Date(), getDay(), getMonth(), getWeekday(), getYear(), getYearday(), Rcpp::operator+(), and update_↩ tm().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Date.h

6.168 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

Private Member Functions

- void update_tm ()

Private Attributes

- double m_dt
- struct tm m_tm
- unsigned int m_us

Friends

- Datetime operator+ (const Datetime &dt, double 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)

6.168.1 Detailed Description

Definition at line 29 of file Datetime.h.

6.168.2 Constructor & Destructor Documentation

6.168.2.1 Rcpp::Datetime::Datetime ( ) [inline]

Definition at line 31 of file Datetime.h.

References m_dt, and update_tm().

Referenced by Datetime().

Here is the call graph for this function:

6.168.2.2 Rcpp::Datetime::Datetime (SEXP *s*) [inline]

Definition at line 25 of file Datetime.h.

References m_dt, and update_tm().

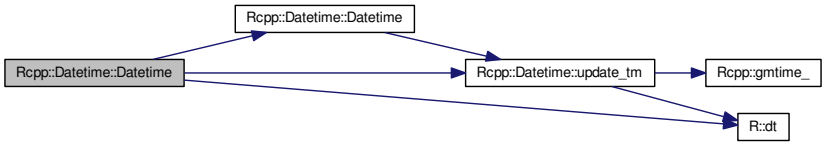
Here is the call graph for this function:

6.168.2.3 Rcpp::Datetime::Datetime (const double & *dt*) [inline]

Definition at line 38 of file Datetime.h.

References Datetime(), R::dt(), m_dt, and update_tm().

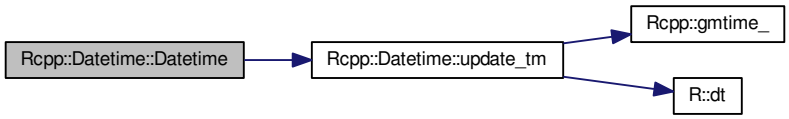
Here is the call graph for this function:



Definition at line 30 of file Datetime.h.

References m_dt, and update_tm().

Here is the call graph for this function:



6.168.3 Member Function Documentation

6.168.3.1 int Rcpp::Datetime::getDay ( ) const [inline]

Definition at line 50 of file Datetime.h.

References m_tm.

6.168.3.2 double Rcpp::Datetime::getFractionalTimestamp (void) const [inline]

Definition at line 44 of file Datetime.h.

References m_dt.

6.168.3.3 int Rcpp::Datetime::getHours ( ) const [inline]

Definition at line 49 of file Datetime.h.

References m_tm.

6.168.3.4 int Rcpp::Datetime::getMicroSeconds ( ) const [inline]

Definition at line 46 of file Datetime.h.

References m_us.

6.168.3.5 int Rcpp::Datetime::getMinutes ( ) const [inline]

Definition at line 48 of file Datetime.h.

References m_tm.

6.168.3.6 int Rcpp::Datetime::getMonth ( ) const [inline]

Definition at line 51 of file Datetime.h.

References m_tm.

6.168.3.7 int Rcpp::Datetime::getSeconds ( ) const [inline]

Definition at line 47 of file Datetime.h.

References m_tm.

6.168.3.8 int Rcpp::Datetime::getWeekday ( ) const [inline]

Definition at line 53 of file Datetime.h.

References m_tm.

6.168.3.9 int Rcpp::Datetime::getYear ( ) const [inline]

Definition at line 52 of file Datetime.h.

References m_tm.

6.168.3.10 int Rcpp::Datetime::getYearday ( ) const [inline]

Definition at line 54 of file Datetime.h.

References R::dt(), m_tm, operator!=, operator+, operator-, operator<, operator<=, operator==, operator>, and operator>=.

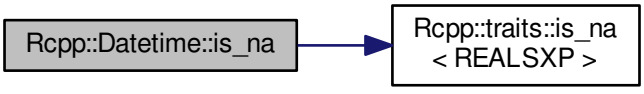
Here is the call graph for this function:

6.168.3.11 int Rcpp::Datetime::is_na ( ) const [inline]

Definition at line 66 of file Datetime.h.

References Rcpp::traits::is_na< REALSXP >(), and m_dt.

Here is the call graph for this function:



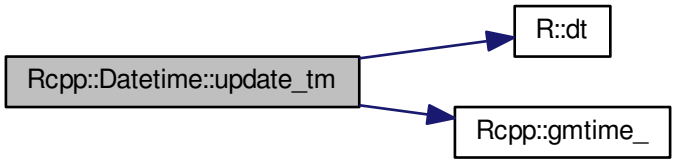
6.168.3.12 void Rcpp::Datetime::update_tm ( ) [inline], [private]

Definition at line 74 of file Datetime.h.

References R::dt(), Rcpp::gmtime_(), and m_tm.

Referenced by Datetime().

Here is the call graph for this function:



6.168.4 Friends And Related Function Documentation

6.168.4.1 bool operator!= (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 121 of file Datetime.h.

Referenced by getYearday().

6.168.4.2 Datetime operator+ (const Datetime & *dt,* double *offset*) [friend]

Definition at line 106 of file Datetime.h.

Referenced by getYearday().

6.168.4.3 double operator- (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 115 of file Datetime.h.

Referenced by getYearday().

6.168.4.4 bool operator< (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 116 of file Datetime.h.

Referenced by getYearday().

6.168.4.5 bool operator<= (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 120 of file Datetime.h.

Referenced by getYearday().

6.168.4.6 bool operator== (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 118 of file Datetime.h.

Referenced by getYearday().

6.168.4.7 bool operator> (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 117 of file Datetime.h.

Referenced by getYearday().

6.168.4.8 bool operator>= (const Datetime & *dt1,* const Datetime & *dt2*) [friend]

Definition at line 119 of file Datetime.h.

Referenced by getYearday().

6.168.5 Member Data Documentation

6.168.5.1 double Rcpp::Datetime::m_dt [private]

Definition at line 69 of file Datetime.h.

Referenced by Datetime(), getFractionalTimestamp(), is_na(), Rcpp::operator!=(), Rcpp::operator+(), Rcpp::operator-(), Rcpp::operator<(), Rcpp::operator<=(), Rcpp::operator==(), Rcpp::operator>(), and Rcpp::operator>=().

6.168.5.2 struct tm Rcpp::Datetime::m_tm [private]

Definition at line 70 of file Datetime.h.

Referenced by getDay(), getHours(), getMinutes(), getMonth(), getSeconds(), getWeekday(), getYear(), getYearday(), Rcpp::operator+(), and update_tm().

6.168.5.3 unsigned int Rcpp::Datetime::m_us [private]

Definition at line 71 of file Datetime.h.

Referenced by getMicroSeconds(), and Rcpp::operator+().

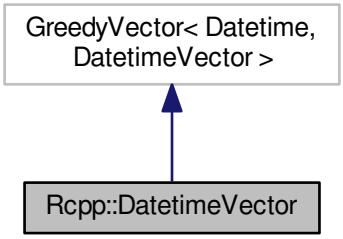
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Datetime.h

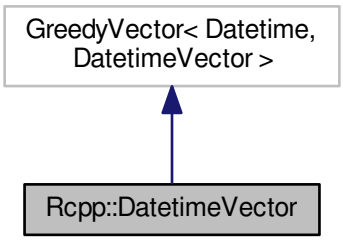
6.169 Rcpp::DatetimeVector Class Reference

`#include <DatetimeVector.h>`

Inheritance diagram for Rcpp::DatetimeVector:



Collaboration diagram for Rcpp::DatetimeVector:



Public Member Functions

- DatetimeVector (SEXP vec)
- DatetimeVector (int n)
- std::vector< Datetime > getDatetimes () const

6.169.1 Detailed Description

Definition at line 31 of file DatetimeVector.h.

6.169.2 Constructor & Destructor Documentation

6.169.2.1 Rcpp::DatetimeVector::DatetimeVector (SEXP *vec*) [inline]

Definition at line 33 of file DatetimeVector.h.

6.169.2.2 Rcpp::DatetimeVector::DatetimeVector (int *n*) [inline]

Definition at line 34 of file DatetimeVector.h.

6.169.3 Member Function Documentation

6.169.3.1 std::vector<Datetime> Rcpp::DatetimeVector::getDatetimes ( ) const [inline]

Definition at line 36 of file DatetimeVector.h.

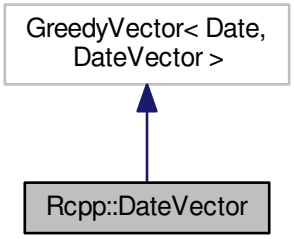
The documentation for this class was generated from the following file:

- inst/include/Rcpp/DatetimeVector.h

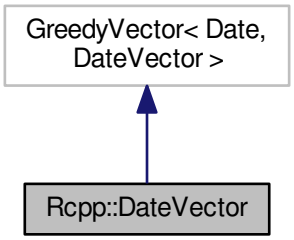
6.170 Rcpp::DateVector Class Reference

`#include <DateVector.h>`

Inheritance diagram for Rcpp::DateVector:



Collaboration diagram for Rcpp::DateVector:



Public Member Functions

- DateVector (SEXP vec)
- DateVector (int n)
- std::vector< Date > getDates () const

6.170.1 Detailed Description

Definition at line 30 of file DateVector.h.

6.170.2 Constructor & Destructor Documentation

6.170.2.1 Rcpp::DateVector::DateVector (SEXP *vec*) [inline]

Definition at line 32 of file DateVector.h.

6.170.2.2 Rcpp::DateVector::DateVector (int *n*) [inline]

Definition at line 33 of file DateVector.h.

6.170.3 Member Function Documentation

6.170.3.1 std::vector<Date> Rcpp::DateVector::getDates ( ) const [inline]

Definition at line 35 of file DateVector.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/DateVector.h

6.171 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

6.171.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::decays_to_ctype< T >

Definition at line 110 of file algorithm.h.

6.171.2 Member Function Documentation

6.171.2.1 template<typename T > static T Rcpp::algorithm::helpers::decays_to_ctype< T >::make ( ) [static]

6.171.2.2 template<typename T > static CTYPE_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const char &) [static]

6.171.2.3 template<typename T > static CTYPE_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const short &) [static]

6.171.2.4 template<typename T > static CTYPE_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const int &) [static]

6.171.2.5 template<typename T > static CTYPE_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const long &) [static]

6.171.2.6 template<typename T > static CTYPE_FLOAT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const float &) [static]

6.171.2.7 template<typename T > static CTYPE_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const double &) [static]

6.171.2.8 template<typename T > static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const long double &) [static]

6.171.2.9 template<typename T > static CTYPE_STRING Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const std::string &) [static]

6.171.2.10 template<typename T > static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const unsigned char &) [static]

6.171.2.11 template<typename T > static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const unsigned short &) [static]

6.171.2.12 template<typename T > static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const unsigned int &) [static]

6.171.2.13 template<typename T > static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (const unsigned long &) [static]

6.171.2.14 template<typename T > static CTYPE_UNKNOWN Rcpp::algorithm::helpers::decays_to_ctype< T >::test (...) [static]

6.171.3 Member Data Documentation

6.171.3.1 template<typename T > const bool Rcpp::algorithm::helpers::decays_to_ctype< T >::value = ctype_helper< sizeof(test(make())) >::value [static]

Definition at line 134 of file algorithm.h.

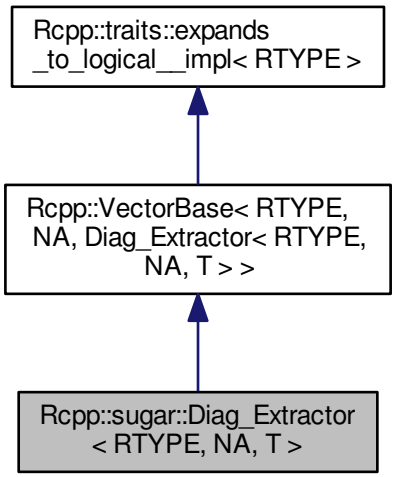
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

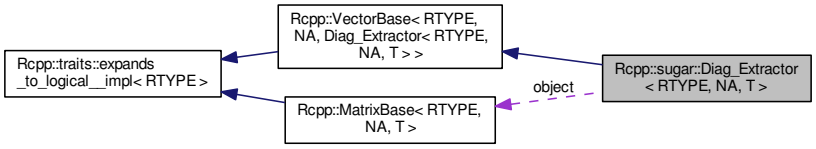
6.172 Rcpp::sugar::Diag_Extractor< RTYPE, NA, T > Class Template Reference

`#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

6.172.1 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.

6.172.2 Member Typedef Documentation

6.172.2.1 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.

6.172.2.2 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.

6.172.3 Constructor & Destructor Documentation

6.172.3.1 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.

6.172.4 Member Function Documentation

6.172.4.1 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.

6.172.4.2 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.

6.172.5 Member Data Documentation

6.172.5.1 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(), Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::ncol(), Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::nrow(), Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::size(), and Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::size().

6.172.5.2 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[]().

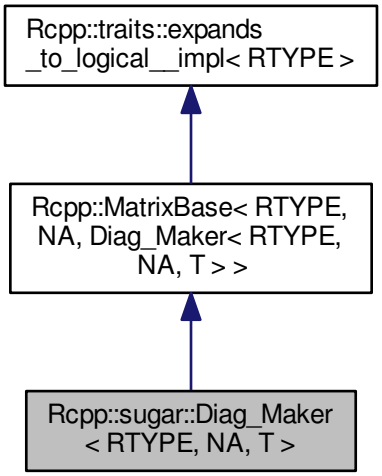
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

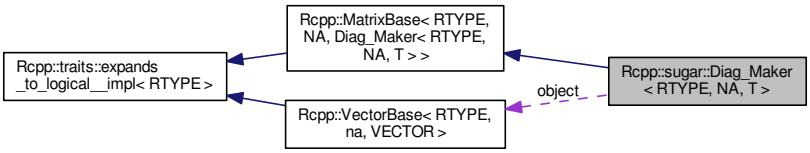
6.173 Rcpp::sugar::Diag_Maker< RTYPE, NA, T > Class Template Reference

`#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

6.173.1 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.

6.173.2 Member Typedef Documentation

6.173.2.1 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.

6.173.2.2 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.

6.173.3 Constructor & Destructor Documentation

6.173.3.1 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.

6.173.4 Member Function Documentation

6.173.4.1 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_Extractor< RTYPE, NA, T >::n.

6.173.4.2 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_Extractor< RTYPE, NA, T >::n.

6.173.4.3 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.

6.173.4.4 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_Extractor< RTYPE, NA, T >::n.

6.173.5 Member Data Documentation

6.173.5.1 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.

6.173.5.2 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

6.174 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

6.174.1 Detailed Description

template<typename T>
struct Rcpp::sugar::diag_result_type_trait< T >

Definition at line 71 of file diag.h.

6.174.2 Member Typedef Documentation

6.174.2.1 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.

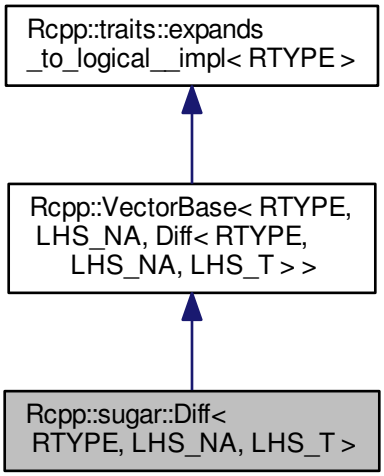
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

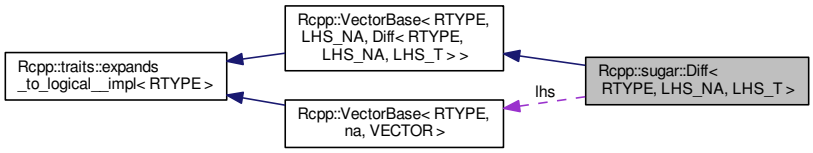
6.175 Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T > Class Template Reference

`#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

6.175.1 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.

6.175.2 Member Typedef Documentation

6.175.2.1 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.

6.175.2.2 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.

6.175.3 Constructor & Destructor Documentation

6.175.3.1 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.

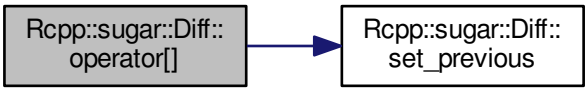
6.175.4 Member Function Documentation

6.175.4.1 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:



6.175.4.2 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[ ]().

6.175.4.3 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:

6.175.5 Member Data Documentation

6.175.5.1 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::diff(), 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().

6.175.5.2 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().

6.175.5.3 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().

6.175.5.4 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().

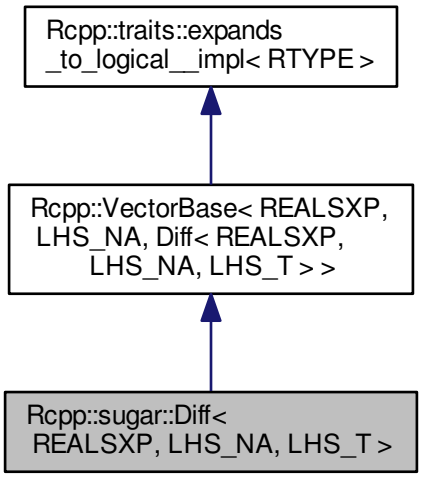
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/diff.h

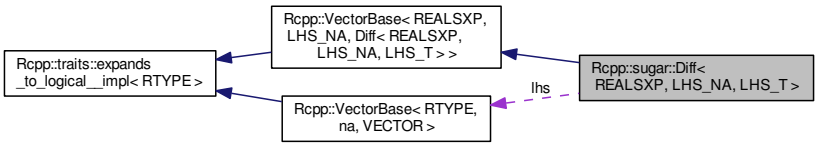
6.176 Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T > Class Template Reference

`#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

6.176.1 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.

6.176.2 Member Typedef Documentation

6.176.2.1 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.

6.176.3 Constructor & Destructor Documentation

6.176.3.1 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.

6.176.4 Member Function Documentation

6.176.4.1 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.

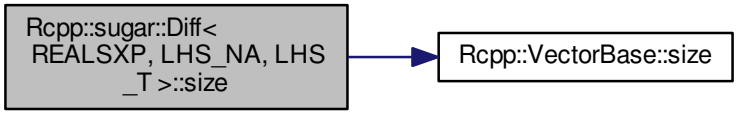
References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous, and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index.

6.176.4.2 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:



6.176.5 Member Data Documentation

6.176.5.1 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.

6.176.5.2 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.

6.176.5.3 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.

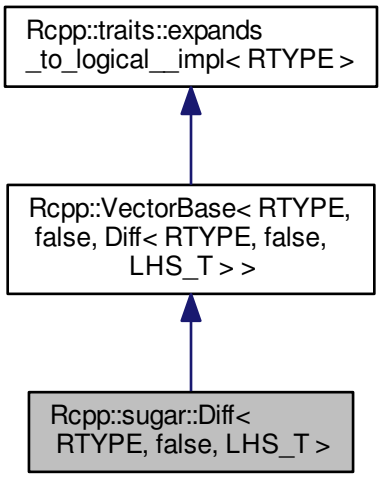
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/diff.h

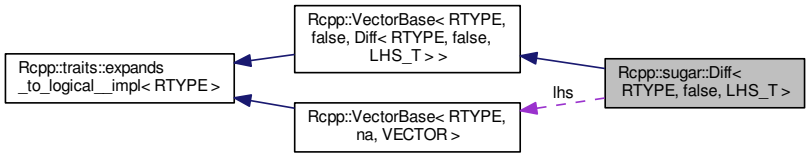
6.177 Rcpp::sugar::Diff< RTYPE, false, LHS_T > Class Template Reference

`#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

6.177.1 Detailed Description

template<int RTYPE, typename LHS_T>
class Rcpp::sugar::Diff< RTYPE, false, LHS_T >

Definition at line 98 of file diff.h.

6.177.2 Member Typedef Documentation

6.177.2.1 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.

6.177.2.2 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.

6.177.3 Constructor & Destructor Documentation

6.177.3.1 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.

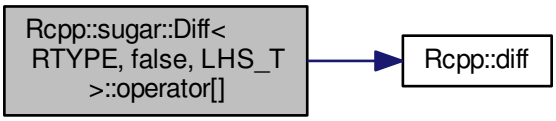
6.177.4 Member Function Documentation

6.177.4.1 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:

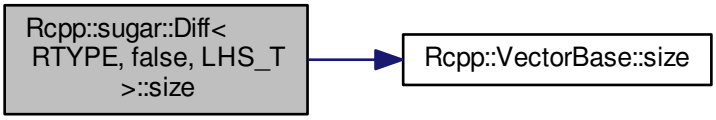


6.177.4.2 template<int RTYPE, typename LHS_T > R_xlen_t Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size ( ) const [inline]

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:



6.177.5 Member Data Documentation

6.177.5.1 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.

6.177.5.2 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.

6.177.5.3 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

6.178 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

6.178.1 Detailed Description

Definition at line 27 of file Dimension.h.

6.178.2 Member Typedef Documentation

6.178.2.1 typedef std::vector<int>::const_reference Rcpp::Dimension::const_reference

Definition at line 30 of file Dimension.h.

6.178.2.2 typedef std::vector<int>::reference Rcpp::Dimension::reference

Definition at line 29 of file Dimension.h.

6.178.3 Constructor & Destructor Documentation

6.178.3.1 Rcpp::Dimension::Dimension ( ) [inline]

Definition at line 32 of file Dimension.h.

References dims.

6.178.3.2 Rcpp::Dimension::Dimension (SEXP *dims*) [inline]

Definition at line 23 of file Dimension.h.

6.178.3.3 Rcpp::Dimension::Dimension (const Dimension & *other*) [inline]

Definition at line 36 of file Dimension.h.

6.178.3.4 Rcpp::Dimension::Dimension (const size_t & *n1*) [inline]

Definition at line 42 of file Dimension.h.

6.178.3.5 Rcpp::Dimension::Dimension (const size_t & *n1*, const size_t & *n2*) [inline]

Definition at line 45 of file Dimension.h.

6.178.3.6 Rcpp::Dimension::Dimension (const size_t & *n1*, const size_t & *n2*, const size_t & *n3*) [inline]

Definition at line 49 of file Dimension.h.

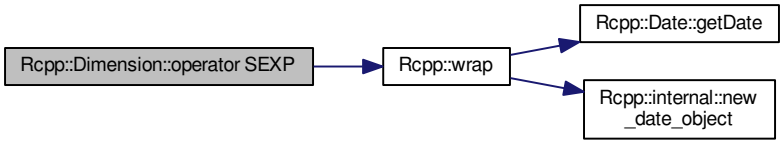
6.178.4 Member Function Documentation

6.178.4.1 Rcpp::Dimension::operator SEXP ( ) const [inline]

Definition at line 25 of file Dimension.h.

References dims, and Rcpp::wrap().

Here is the call graph for this function:



6.178.4.2 Dimension& Rcpp::Dimension::operator= (const Dimension & *other*) [inline]

Definition at line 37 of file Dimension.h.

References dims.

6.178.4.3 reference Rcpp::Dimension::operator[ ] (int *i*) [inline]

Definition at line 63 of file Dimension.h.

6.178.4.4 const_reference Rcpp::Dimension::operator[ ] (int *i*) const [inline]

Definition at line 67 of file Dimension.h.

6.178.4.5 R_xlen_t Rcpp::Dimension::prod ( ) const [inline]

Definition at line 59 of file Dimension.h.

Referenced by Rcpp::Vector< INTSXP >::Vector().

6.178.4.6 int Rcpp::Dimension::size ( ) const [inline]

Definition at line 56 of file Dimension.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Vector< INTSXP >::Vector().

6.178.5 Member Data Documentation

6.178.5.1 std::vector<int> Rcpp::Dimension::dims [private]

Definition at line 73 of file Dimension.h.

Referenced by Dimension(), operator SEXP(), and operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Dimension.h

6.179 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_

6.179.1 Detailed Description

Definition at line 28 of file DimNameProxy.h.

6.179.2 Constructor & Destructor Documentation

6.179.2.1 Rcpp::internal::DimNameProxy::DimNameProxy (SEXP *data,* int *dim*) [inline]

Definition at line 32 of file DimNameProxy.h.

6.179.2.2 Rcpp::internal::DimNameProxy::DimNameProxy (DimNameProxy const & *other*) [inline]

Definition at line 33 of file DimNameProxy.h.

6.179.3 Member Function Documentation

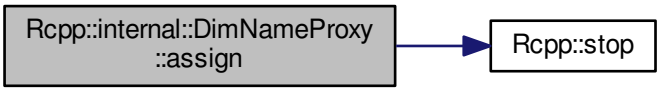
6.179.3.1 DimNameProxy& Rcpp::internal::DimNameProxy::assign (SEXP *other*) [inline]

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:



6.179.3.2 Rcpp::internal::DimNameProxy::operator SEXP ( ) const [inline]

Definition at line 66 of file DimNameProxy.h.

References data_, and dim_.

6.179.3.3 template<typename T > Rcpp::internal::DimNameProxy::operator T ( ) const [inline]

Definition at line 72 of file DimNameProxy.h.

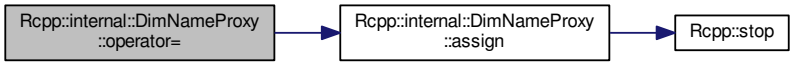
References data_, and dim_.

6.179.3.4 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:

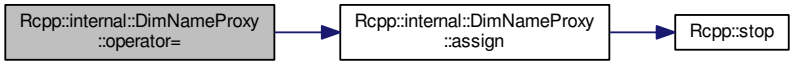


6.179.3.5 DimNameProxy& Rcpp::internal::DimNameProxy::operator= (const DimNameProxy & *other*) [inline]

Definition at line 62 of file DimNameProxy.h.

References assign().

Here is the call graph for this function:



6.179.4 Member Data Documentation

6.179.4.1 SEXP Rcpp::internal::DimNameProxy::data_ [private]

Definition at line 83 of file DimNameProxy.h.

Referenced by assign(), operator SEXP(), and operator T().

6.179.4.2 int Rcpp::internal::DimNameProxy::dim_ [private]

Definition at line 84 of file DimNameProxy.h.

Referenced by assign(), operator SEXP(), and operator T().

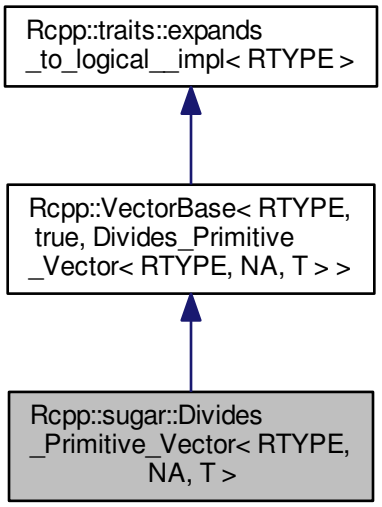
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/DimNameProxy.h

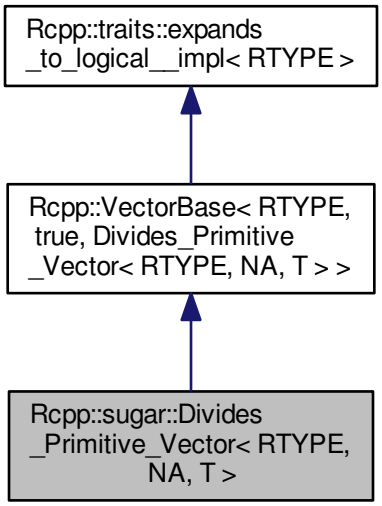
6.180 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T > Class Template Reference

```
#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

6.180.1 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.

6.180.2 Member Typedef Documentation

6.180.2.1 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.

6.180.2.2 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.

6.180.2.3 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.

6.180.3 Constructor & Destructor Documentation

6.180.3.1 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.

6.180.4 Member Function Documentation

6.180.4.1 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_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.

6.180.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.180.5 Member Data Documentation

6.180.5.1 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.

6.180.5.2 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.

6.180.5.3 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.

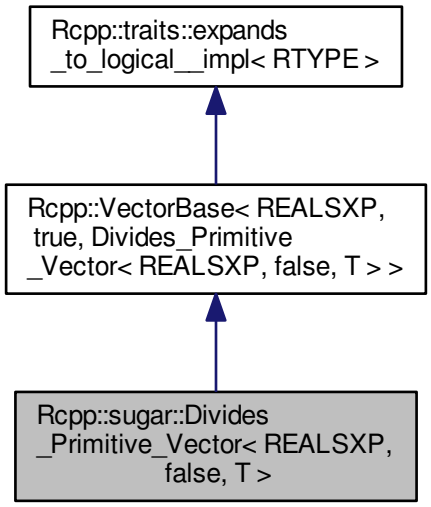
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

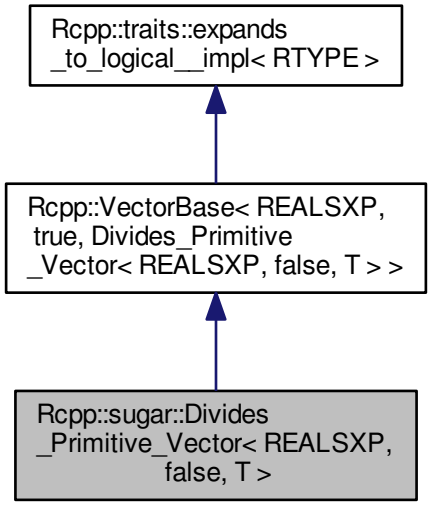
6.181 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T > Class Template Reference

`#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

6.181.1 Detailed Description

template<typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >

Definition at line 389 of file divides.h.

6.181.2 Member Typedef Documentation

6.181.2.1 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.

6.181.2.2 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.

6.181.3 Constructor & Destructor Documentation

6.181.3.1 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.

6.181.4 Member Function Documentation

6.181.4.1 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_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.

6.181.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.181.5 Member Data Documentation

6.181.5.1 template<typename T > double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::lhs [private]

Definition at line 404 of file divides.h.

6.181.5.2 template<typename T > const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::rhs [private]

Definition at line 405 of file divides.h.

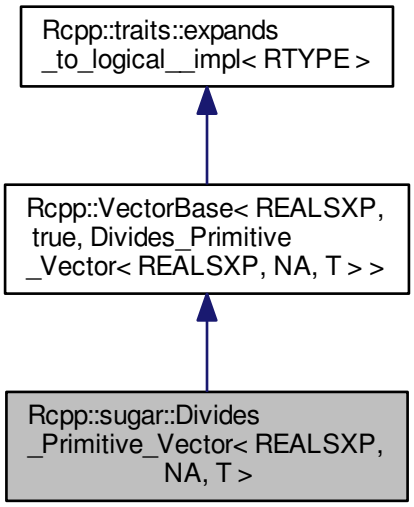
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

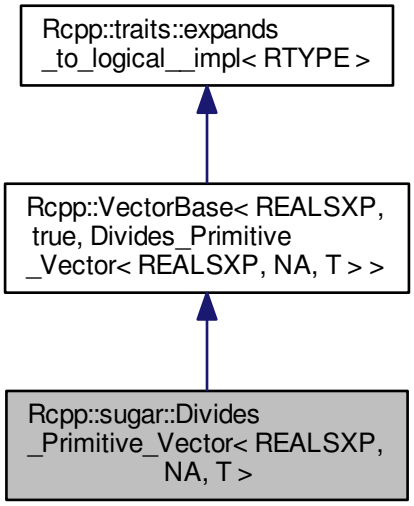
6.182 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T > Class Template Reference

`#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

6.182.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >

Definition at line 345 of file divides.h.

6.182.2 Member Typedef Documentation

6.182.2.1 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.

6.182.2.2 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.

6.182.3 Constructor & Destructor Documentation

6.182.3.1 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.

6.182.4 Member Function Documentation

6.182.4.1 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_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.

6.182.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.182.5 Member Data Documentation

6.182.5.1 template<bool NA, typename T > double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::lhs [private]

Definition at line 359 of file divides.h.

6.182.5.2 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.

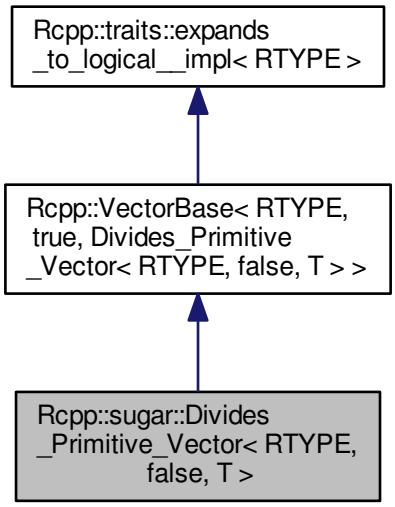
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

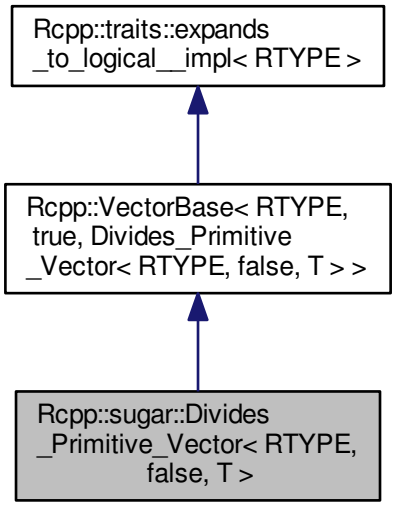
6.183 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T > Class Template Reference

```
#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

6.183.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >

Definition at line 366 of file divides.h.

6.183.2 Member Typedef Documentation

6.183.2.1 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.

6.183.2.2 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.

6.183.2.3 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.

6.183.3 Constructor & Destructor Documentation

6.183.3.1 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.

6.183.4 Member Function Documentation

6.183.4.1 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_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.

6.183.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.183.5 Member Data Documentation

6.183.5.1 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::lhs [private]

Definition at line 383 of file divides.h.

6.183.5.2 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.

6.183.5.3 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.

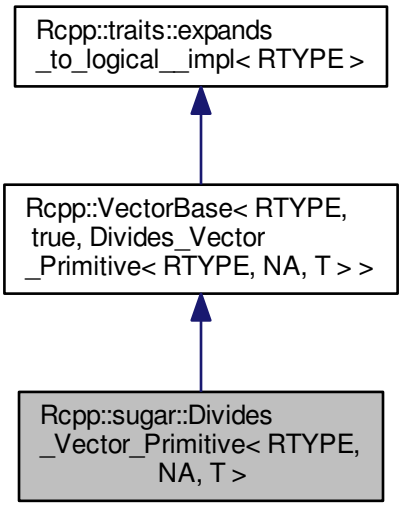
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

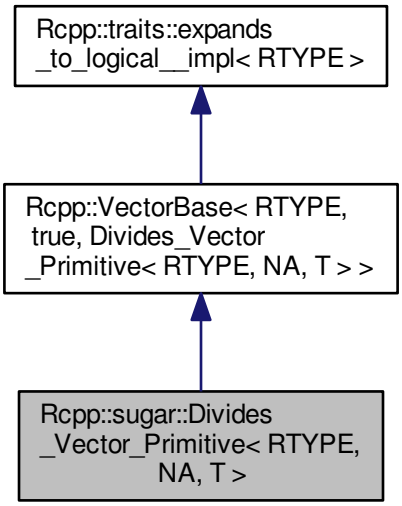
6.184 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T > Class Template Reference

`#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

6.184.1 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.

6.184.2 Member Typedef Documentation

6.184.2.1 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.

6.184.2.2 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.

6.184.2.3 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.

6.184.3 Constructor & Destructor Documentation

6.184.3.1 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.

6.184.4 Member Function Documentation

6.184.4.1 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_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.

6.184.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.184.5 Member Data Documentation

6.184.5.1 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.

6.184.5.2 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.

6.184.5.3 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.

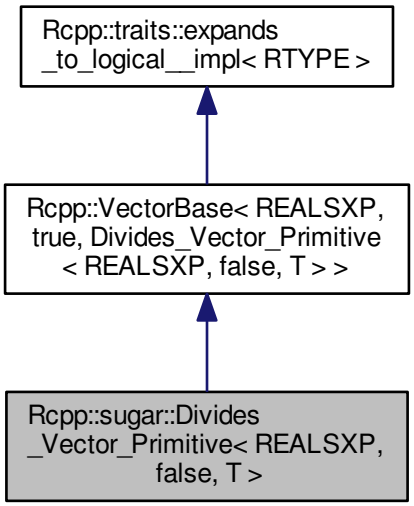
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

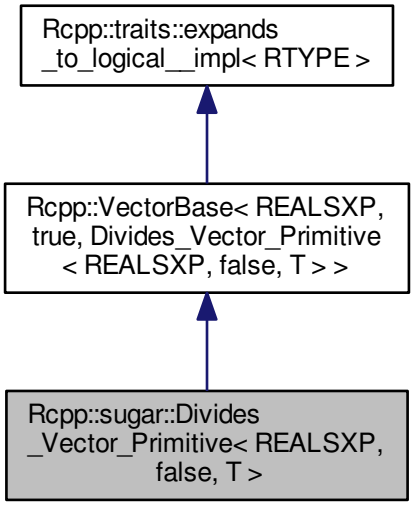
6.185 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T > Class Template Reference

`#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

6.185.1 Detailed Description

template<typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >

Definition at line 300 of file divides.h.

6.185.2 Member Typedef Documentation

6.185.2.1 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.

6.185.2.2 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.

6.185.3 Constructor & Destructor Documentation

6.185.3.1 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.

6.185.4 Member Function Documentation

6.185.4.1 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_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.

6.185.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.185.5 Member Data Documentation

6.185.5.1 template<typename T > const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::lhs [private]

Definition at line 315 of file divides.h.

6.185.5.2 template<typename T > double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::rhs [private]

Definition at line 316 of file divides.h.

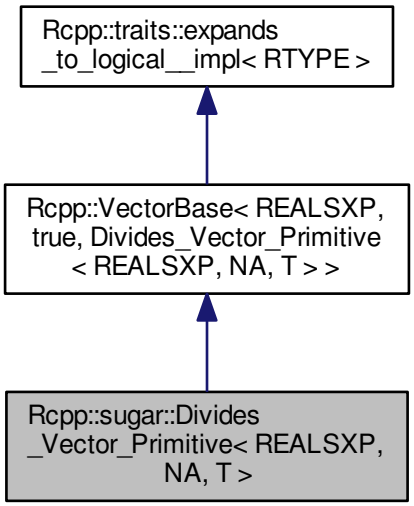
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

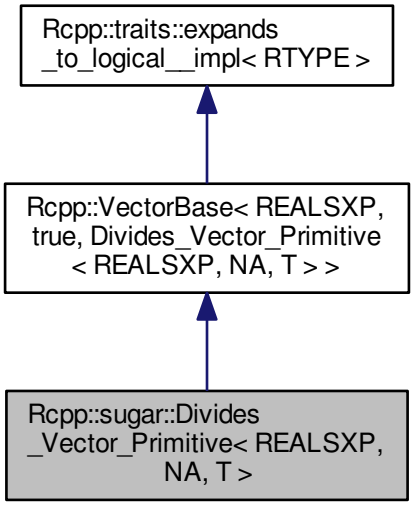
6.186 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T > Class Template Reference

`#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

6.186.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >

Definition at line 252 of file divides.h.

6.186.2 Member Typedef Documentation

6.186.2.1 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.

6.186.2.2 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.

6.186.3 Constructor & Destructor Documentation

6.186.3.1 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.

6.186.4 Member Function Documentation

6.186.4.1 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_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.

6.186.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.186.5 Member Data Documentation

6.186.5.1 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.

6.186.5.2 template<bool NA, typename T > double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::rhs [private]

Definition at line 270 of file divides.h.

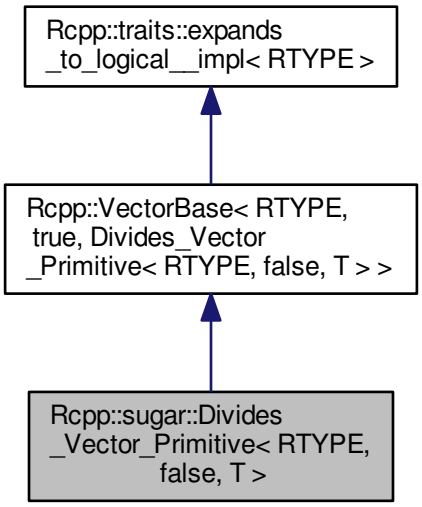
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

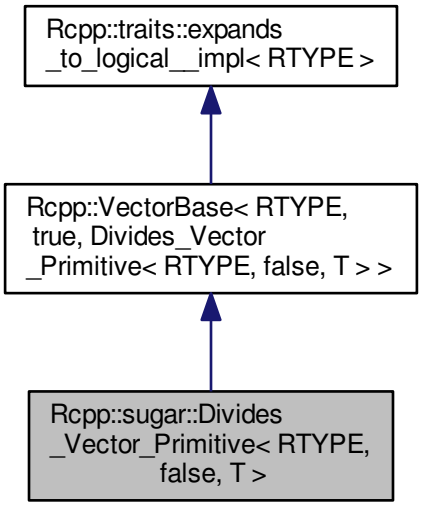
6.187 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T > Class Template Reference

`#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

6.187.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >

Definition at line 276 of file divides.h.

6.187.2 Member Typedef Documentation

6.187.2.1 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.

6.187.2.2 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.

6.187.2.3 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.

6.187.3 Constructor & Destructor Documentation

6.187.3.1 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.

6.187.4 Member Function Documentation

6.187.4.1 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_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.

6.187.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.187.5 Member Data Documentation

6.187.5.1 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.

6.187.5.2 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::rhs [private]

Definition at line 295 of file divides.h.

6.187.5.3 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.

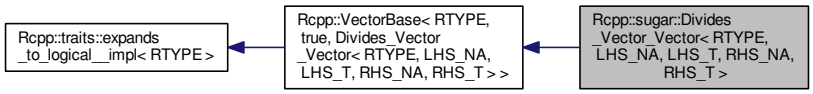
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

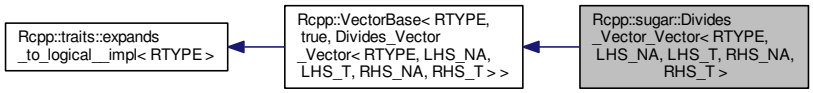
6.188 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.188.1 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.

6.188.2 Member Typedef Documentation

6.188.2.1 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.

6.188.2.2 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.

6.188.2.3 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.

6.188.2.4 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.

6.188.2.5 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.

6.188.3 Constructor & Destructor Documentation

6.188.3.1 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.

6.188.4 Member Function Documentation

6.188.4.1 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.

6.188.4.2 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.

6.188.5 Member Data Documentation

6.188.5.1 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::operator/(), 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_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_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_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(), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size(), 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().

6.188.5.2 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::operator/(), 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_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_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< REALSXP, false, T >::size().

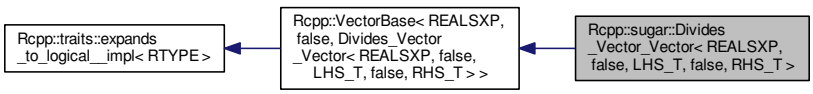
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

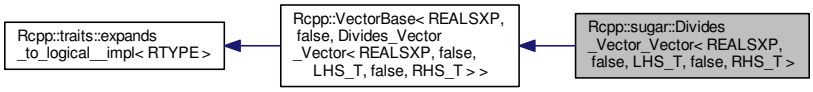
6.189 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.189.1 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.

6.189.2 Member Typedef Documentation

6.189.2.1 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.

6.189.2.2 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.

6.189.2.3 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.

6.189.2.4 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.

6.189.3 Constructor & Destructor Documentation

6.189.3.1 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.

6.189.4 Member Function Documentation

6.189.4.1 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.

6.189.4.2 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.

6.189.5 Member Data Documentation

6.189.5.1 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.

6.189.5.2 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.

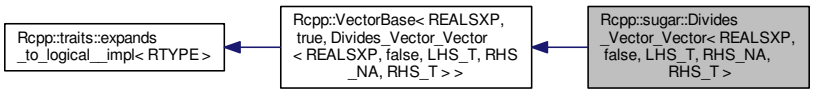
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

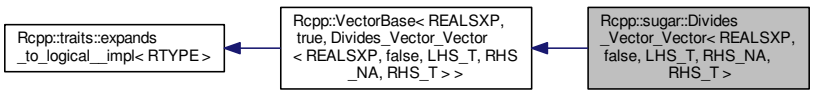
6.190 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.190.1 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.

6.190.2 Member Typedef Documentation

6.190.2.1 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.

6.190.2.2 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.

6.190.2.3 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.

6.190.2.4 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.

6.190.3 Constructor & Destructor Documentation

6.190.3.1 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.

6.190.4 Member Function Documentation

6.190.4.1 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.

6.190.4.2 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.

6.190.5 Member Data Documentation

6.190.5.1 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.

6.190.5.2 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.

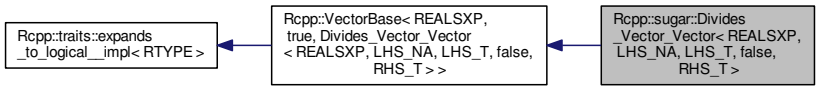
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

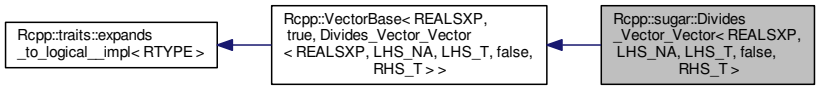
6.191 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.191.1 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.

6.191.2 Member Typedef Documentation

6.191.2.1 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.

6.191.2.2 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.

6.191.2.3 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.

6.191.2.4 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.

6.191.3 Constructor & Destructor Documentation

6.191.3.1 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.

6.191.4 Member Function Documentation

6.191.4.1 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.

6.191.4.2 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.

6.191.5 Member Data Documentation

6.191.5.1 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.

6.191.5.2 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.

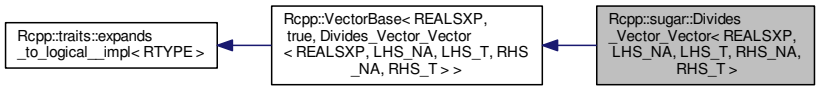
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

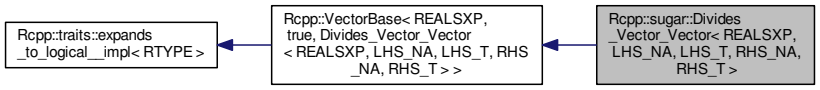
6.192 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.192.1 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.

6.192.2 Member Typedef Documentation

6.192.2.1 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.

6.192.2.2 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.

6.192.2.3 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.

6.192.2.4 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.

6.192.3 Constructor & Destructor Documentation

6.192.3.1 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.

6.192.4 Member Function Documentation

6.192.4.1 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.

6.192.4.2 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.

6.192.5 Member Data Documentation

6.192.5.1 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.

6.192.5.2 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.

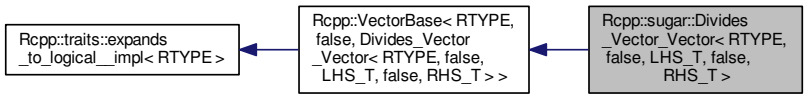
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

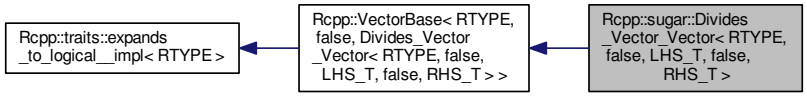
6.193 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.193.1 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.

6.193.2 Member Typedef Documentation

6.193.2.1 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.

6.193.2.2 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.

6.193.2.3 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.

6.193.2.4 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.

6.193.2.5 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.

6.193.3 Constructor & Destructor Documentation

6.193.3.1 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.

6.193.4 Member Function Documentation

6.193.4.1 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.

6.193.4.2 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.

6.193.5 Member Data Documentation

6.193.5.1 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.

6.193.5.2 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.

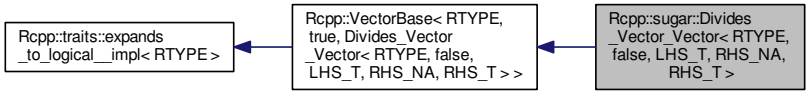
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

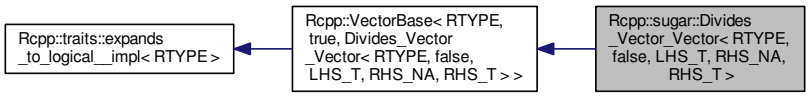
6.194 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.194.1 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.

6.194.2 Member Typedef Documentation

6.194.2.1 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.

6.194.2.2 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.194.2.3 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.

6.194.2.4 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.194.2.5 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.

6.194.3 Constructor & Destructor Documentation

6.194.3.1 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.

6.194.4 Member Function Documentation

6.194.4.1 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.

6.194.4.2 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.

6.194.5 Member Data Documentation

6.194.5.1 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.

6.194.5.2 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.

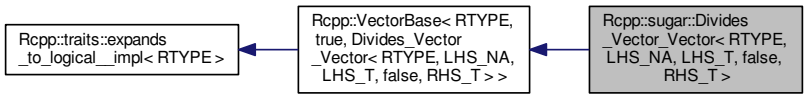
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

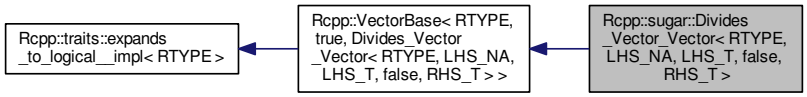
6.195 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.195.1 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.

6.195.2 Member Typedef Documentation

6.195.2.1 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.

6.195.2.2 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.

6.195.2.3 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.

6.195.2.4 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.

6.195.2.5 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.

6.195.3 Constructor & Destructor Documentation

6.195.3.1 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.

6.195.4 Member Function Documentation

6.195.4.1 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.

6.195.4.2 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.

6.195.5 Member Data Documentation

6.195.5.1 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.

6.195.5.2 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

6.196 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)

6.196.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairImpl< CLASS >

Definition at line 24 of file DottedPairImpl.h.

6.196.2 Member Function Documentation

6.196.2.1 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

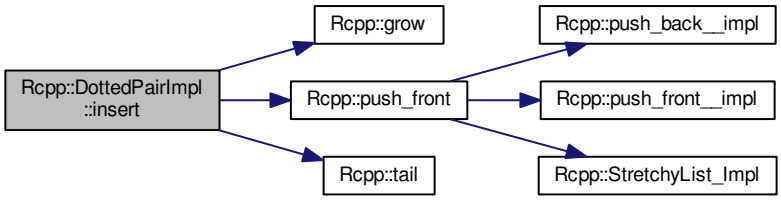
Parameters

index	index (0-based) where to insert
object	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:



6.196.2.2 template<typename CLASS > R_xlen_t Rcpp::DottedPairImpl< CLASS >::length ( ) const [inline]

Definition at line 66 of file DottedPairImpl.h.

6.196.2.3 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)

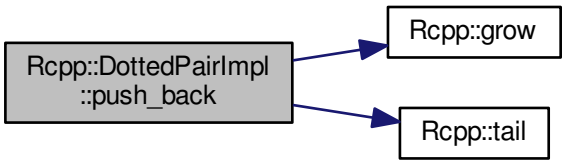
Parameters

object	anything that can be wrapped by one of the wrap functions, named objects (instances of traits::named_object<> 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:



6.196.2.4 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.

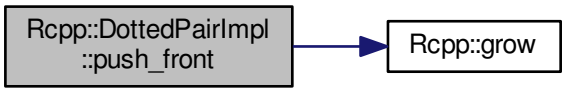
Parameters

object	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:



6.196.2.5 template<typename CLASS > void Rcpp::DottedPairImpl< CLASS >::remove (const size_t & *index*)

Remove the element at the given position

Parameters

index	position where the element is to be removed

Definition at line 85 of file DottedPairImpl.h.

6.196.2.6 template<typename CLASS > template<typename T > void Rcpp::DottedPairImpl< CLASS >::replace (const int & *index*, const T & *object*)

replaces an element of the list

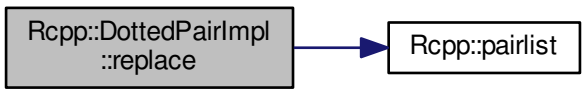
Parameters

index	position
object	object that can be wrapped

Definition at line 71 of file DottedPairImpl.h.

References Rcpp::pairlist().

Here is the call graph for this function:



6.196.2.7 template<typename CLASS > R_xlen_t Rcpp::DottedPairImpl< CLASS >::size ( ) const [inline]

Definition at line 70 of file DottedPairImpl.h.

6.196.3 Friends And Related Function Documentation

6.196.3.1 template<typename CLASS > template<typename T > DottedPairImpl& operator<< (DottedPairImpl< CLASS > & *os*, const T & *t*) [friend]

Definition at line 82 of file DottedPairImpl.h.

6.196.3.2 template<typename CLASS > template<typename T > DottedPairImpl& operator>> (const T & *t*, DottedPairImpl< CLASS > & *s*) [friend]

Definition at line 88 of file DottedPairImpl.h.

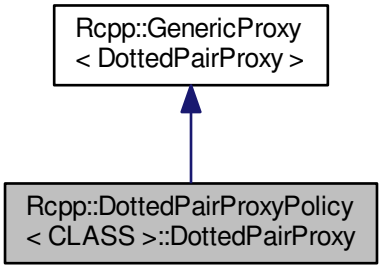
The documentation for this class was generated from the following file:

- inst/include/Rcpp/DottedPairImpl.h

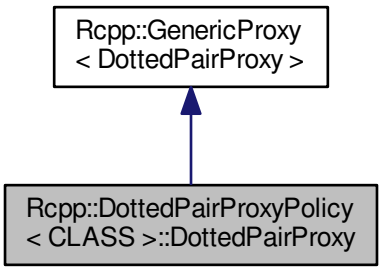
6.197 Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy Class Reference

`#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

6.197.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy

Definition at line 27 of file DottedPairProxy.h.

6.197.2 Constructor & Destructor Documentation

6.197.2.1 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.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::operator[]().

6.197.3 Member Function Documentation

6.197.3.1 template<typename CLASS > SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get ( ) const [inline]

Definition at line 51 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

6.197.3.2 template<typename CLASS > Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator SEXP ( ) const [inline]

Definition at line 54 of file DottedPairProxy.h.

6.197.3.3 template<typename CLASS > template<typename T > Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator T ( ) const

Definition at line 165 of file proxy.h.

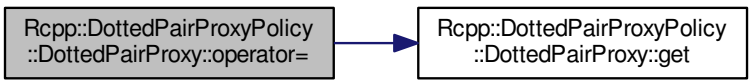
6.197.3.4 template<typename CLASS > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (const DottedPairProxy & *rhs*) [inline]

Definition at line 36 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get().

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

Here is the call graph for this function:



6.197.3.5 template<typename CLASS > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (SEXP *rhs*) [inline]

Definition at line 39 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

Here is the call graph for this function:

6.197.3.6 template<typename CLASS > template<typename T > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (const T & *rhs*)

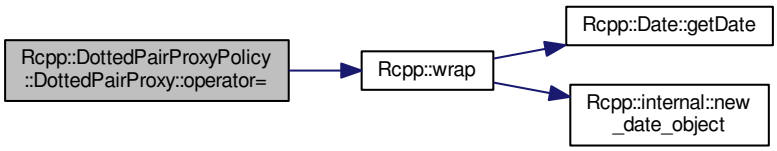
6.197.3.7 template<typename CLASS > template<typename T > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (const traits::named_object< T > & *rhs*)

6.197.3.8 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:

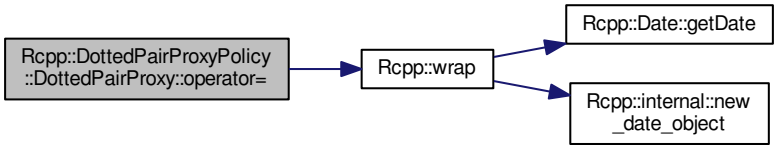


6.197.3.9 template<typename CLASS > template<typename T > DottedPairProxyPolicy<CLASS>::DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (const traits::named_object< T > & *rhs*)

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:



6.197.3.10 template<typename CLASS > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (SEXP *x*) [inline]

Definition at line 57 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.197.3.11 template<typename CLASS > DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (SEXP *x*, const char * *name*) [inline]

Definition at line 61 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.197.4 Member Data Documentation

6.197.4.1 template<typename CLASS > SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node [private]

Definition at line 69 of file DottedPairProxy.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::const_DottedPairProxy(), Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy(), Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get(), Rcpp::DottedPairProxyPolicy< CLASS >::const_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

6.198 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

6.198.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >

Definition at line 24 of file DottedPairProxy.h.

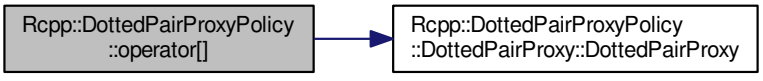
6.198.2 Member Function Documentation

6.198.2.1 template<typename CLASS > DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[ ] (int *i*) [inline]

Definition at line 95 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy().

Here is the call graph for this function:



6.198.2.2 template<typename CLASS > const_DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[ ] (int *i*) const [inline]

Definition at line 98 of file DottedPairProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/DottedPairProxy.h

6.199 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)

6.199.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::element_converter< RTYPE >

Definition at line 28 of file converter.h.

6.199.2 Member Typedef Documentation

6.199.2.1 template<int RTYPE> typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::element_converter< RTYPE >::target

Definition at line 30 of file converter.h.

6.199.3 Member Function Documentation

6.199.3.1 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< INTSXP >::fill__dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), Rcpp::Vector< INTSXP >::insert(), Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()(), Rcpp::Vector< INTSXP >::push_back(), Rcpp::Vector< INTSXP >::push_front(), Rcpp::Vector< INTSXP >::replace_element__dispatch(), and Rcpp::Vector< INTSXP >::replace_element__dispatch__isArgument().

6.199.3.2 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

6.200 Rcpp::traits::enable_if< B, T > Struct Template Reference

`#include <enable_if.h>`

6.200.1 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

6.201 Rcpp::traits::enable_if< true, T > Struct Template Reference

`#include <enable_if.h>`

Public Types

- typedef T type

6.201.1 Detailed Description

template<typename T>
struct Rcpp::traits::enable_if< true, T >

Definition at line 11 of file enable_if.h.

6.201.2 Member Typedef Documentation

6.201.2.1 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

6.202 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

6.202.1 Detailed Description

template<typename Enum, typename Parent>
class Rcpp::enum_< Enum, Parent >

Definition at line 343 of file Module.h.

6.202.2 Member Typedef Documentation

6.202.2.1 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.

6.202.2.2 template<typename Enum , typename Parent > typedef MAP::value_type Rcpp::enum_< Enum, Parent >::PAIR [private]

Definition at line 364 of file Module.h.

6.202.2.3 template<typename Enum , typename Parent > typedef enum_<Enum,Parent> Rcpp::enum_< Enum, Parent >::self

Definition at line 345 of file Module.h.

6.202.3 Constructor & Destructor Documentation

6.202.3.1 template<typename Enum , typename Parent > Rcpp::enum_< Enum, Parent >::enum_ (const char * *name_*) [inline]

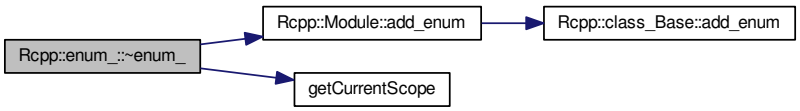
Definition at line 347 of file Module.h.

6.202.3.2 template<typename Enum , typename Parent > Rcpp::enum_< Enum, Parent >::~enum_ ( ) [inline]

Definition at line 350 of file Module.h.

References Rcpp::Module::add_enum(), and getCurrentScope().

Here is the call graph for this function:



6.202.4 Member Function Documentation

6.202.4.1 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.

6.202.5 Member Data Documentation

6.202.5.1 template<typename Enum , typename Parent > std::string Rcpp::enum_< Enum, Parent >::name [private]

Definition at line 362 of file Module.h.

6.202.5.2 template<typename Enum , typename Parent > std::string Rcpp::enum_< Enum, Parent >::parent_typeinfo_name [private]

Definition at line 366 of file Module.h.

6.202.5.3 template<typename Enum , typename Parent > MAP Rcpp::enum_< Enum, Parent >::values [private]

Definition at line 365 of file Module.h.

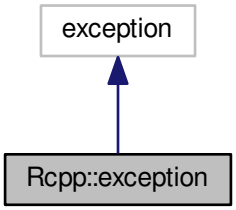
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

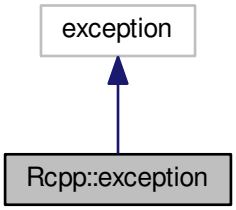
6.203 Rcpp::exception Class Reference

`#include <exceptions.h>`

Inheritance diagram for Rcpp::exception:



Collaboration diagram for Rcpp::exception:



Public Member Functions

- exception (const char *message_)
- exception (const char *message_, const char *file, int line)
- virtual ~exception () throw ()
- virtual const char * what () const throw ()

Private Attributes

- std::string message

6.203.1 Detailed Description

Definition at line 29 of file exceptions.h.

6.203.2 Constructor & Destructor Documentation

6.203.2.1 Rcpp::exception::exception (const char * *message_*) [inline], [explicit]

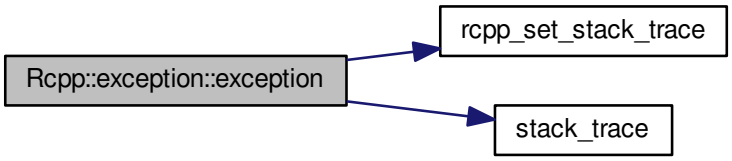
Definition at line 31 of file exceptions.h.

6.203.2.2 Rcpp::exception::exception (const char * *message_*, const char * *file*, int *line*) [inline]

Definition at line 32 of file exceptions.h.

References rcpp_set_stack_trace(), and stack_trace().

Here is the call graph for this function:



6.203.2.3 virtual Rcpp::exception::~exception ( ) throw) [inline], [virtual]

Definition at line 35 of file exceptions.h.

6.203.3 Member Function Documentation

6.203.3.1 virtual const char* Rcpp::exception::what ( ) const throw) [inline], [virtual]

Definition at line 36 of file exceptions.h.

References message.

6.203.4 Member Data Documentation

6.203.4.1 std::string Rcpp::exception::message [private]

Definition at line 40 of file exceptions.h.

Referenced by what(), Rcpp::no_such_env::what(), and Rcpp::file_io_error::what().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

6.204 Rcpp::algorithm::helpers::exp Struct Reference

`#include <algorithm.h>`

Public Member Functions

- template<typename T >
  double operator() (T val)

6.204.1 Detailed Description

Definition at line 187 of file algorithm.h.

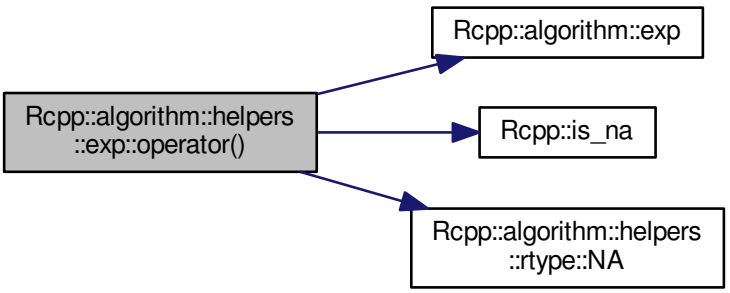
6.204.2 Member Function Documentation

6.204.2.1 template<typename T > double Rcpp::algorithm::helpers::exp::operator() (T *val*) [inline]

Definition at line 189 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:



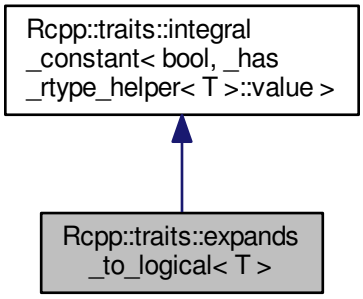
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

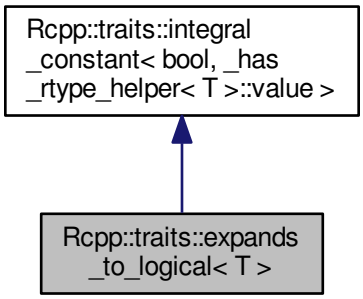
6.205 Rcpp::traits::expands_to_logical< T > Struct Template Reference

`#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

6.205.1 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

6.206 Rcpp::traits::expands_to_logical__impl< RTYPE > Struct Template Reference

`#include <expands_to_logical.h>`

Inherited by Rcpp::MatrixBase< INTSXP, false, Col< RTYPE, LHS_NA, LHS_T > >, Rcpp::MatrixBase< INTSXP, false, Row< RTYPE, LHS_NA, LHS_T > >, 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 > >, Rcpp::MatrixBase< RTYPE, NA, Diag_Maker< RTYPE, NA, T > >, Rcpp::MatrixBase< RTYPE, NA, T >, Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >, Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, ScalarBindable< scalar< RTYPE >::type > > >, Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >, Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, ScalarBindable< scalar< RTYPE >::type > > >, Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, StoragePolicy > >, Rcpp::MatrixBase< RTYPE, true, SubMatrix< RTYPE > >, Rcpp::MatrixBase< RTYPE, na, MATRIX >, Rcpp::VectorBase< RTYPE, na, VECTOR >, Rcpp::VectorBase< INTSXP, false, Range >, Rcpp::VectorBase< INTSXP, false, SeqLen >, Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > >, Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< LGLSXP, false, IsFinite< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsInfinite< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, false, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsNa_Vector_is_na< T > >, Rcpp::VectorBase< LGLSXP, false, IsNaN< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, LowerTri< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< LGLSXP, false, Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< LGLSXP, false, UpperTri< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > >, 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< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< LGLSXP, true, Comparator_With_One_Value< RTYPE, Operator, NA, T > >, Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > >, 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< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sapply< RTYPE, NA, T, Function, NO_CONVERSION > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sapply< RTYPE, NA, T, Function, true > >, 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 > >, 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 > >, 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 > >, 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 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, false, SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_1< NA, RESULT_TYPE, U1, T1 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarComplex< NA, RESULT_TYPE, T, FunPtr > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< T >::rtype, true, Rep_Single< T > >, Rcpp::VectorBase< REALSXP, false, Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive_nona< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, false, Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, false, Pow< INTSXP, false, T, EXPONENT_TYPE > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive_nona< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, false, Vectorized_INTSXP< Func, false, VEC > >, Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > >, Rcpp::VectorBase< REALSXP, NA, D0< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D1< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D3< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P1< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P3< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Pow< INTSXP, NA, T, EXPONENT_TYPE > >, Rcpp::VectorBase< REALSXP, NA, Pow< RTYPE, NA, T, EXPONENT_TYPE > >, Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Q1< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Q2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Q3< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Vectorized< Func, NA, VEC > >, Rcpp::VectorBase< REALSXP, NA, Vectorized_INTSXP< Func, NA, VEC > >, Rcpp::VectorBase< REALSXP, true, Divides_Primitive_Vector< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Primitive_Vector< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive_nona< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive_nona< REALSXP, NA, T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, false, Diff< RTYPE, false, LHS_T > >, Rcpp::VectorBase< RTYPE, false, Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, false, Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

>, Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > >, Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, VECTOR > >, Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive_nona< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, false, Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive_nona< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, NA, Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Diag_Extractor< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Head< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, RangeIndexer< RTYPE, NA, VECTOR > >, Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Rep_each< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Rep_len< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Rev< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Tail< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Comparator_With_One_Value< LGLSXP, Operator, false, T > >, Rcpp::VectorBase< RTYPE, true, ConstMatrixColumn< RTYPE > >, Rcpp::VectorBase< RTYPE, true, ConstMatrixRow< RTYPE > >, Rcpp::VectorBase< RTYPE, true, Divides_Primitive_Vector< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Divides_Primitive_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, false, LHS_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< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, false, COND_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, true, MatrixColumn< RTYPE > >, Rcpp::VectorBase< RTYPE, true, MatrixRow< RTYPE > >, Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive_nona< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, true, Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive_nona< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >, Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, StoragePolicy > >, 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< RTYPE,(LHS_NA||RHS_NA), IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< unary_minus_result_type< RTYPE >::value, NA, UnaryMinus_Vector< unary_minus_result_type< RTYPE >::value, NA, T > >, and Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > >.

6.206.1 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

6.207 Rcpp::traits::expands_to_logical__impl< LGLSXP > Struct Template Reference

`#include <expands_to_logical.h>`

Classes

- struct r_expands_to_logical

6.207.1 Detailed Description

template<>
struct Rcpp::traits::expands_to_logical__impl< LGLSXP >

Definition at line 36 of file expands_to_logical.h.

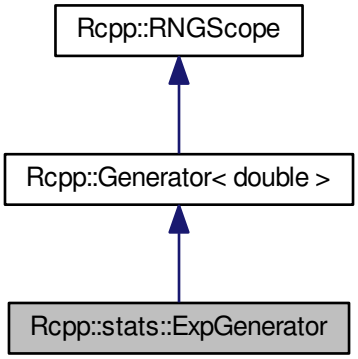
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

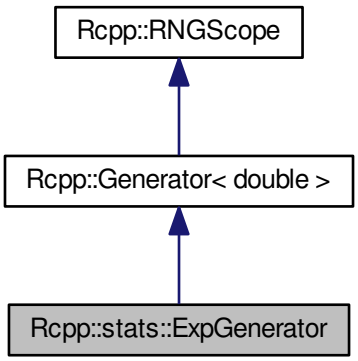
6.208 Rcpp::stats::ExpGenerator Class Reference

`#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

6.208.1 Detailed Description

Definition at line 28 of file rexp.h.

6.208.2 Constructor & Destructor Documentation

6.208.2.1 Rcpp::stats::ExpGenerator::ExpGenerator (double *scale_*) [inline]

Definition at line 31 of file rexp.h.

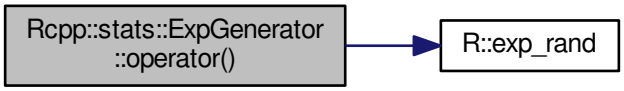
6.208.3 Member Function Documentation

6.208.3.1 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:



6.208.4 Member Data Documentation

6.208.4.1 double Rcpp::stats::ExpGenerator::scale [private]

Definition at line 38 of file rexp.h.

Referenced by operator()().

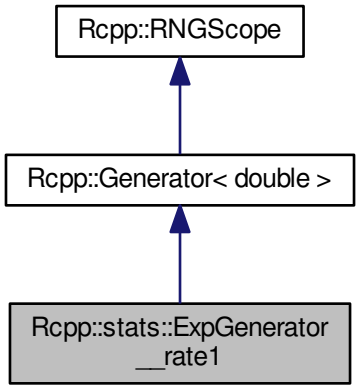
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rexp.h

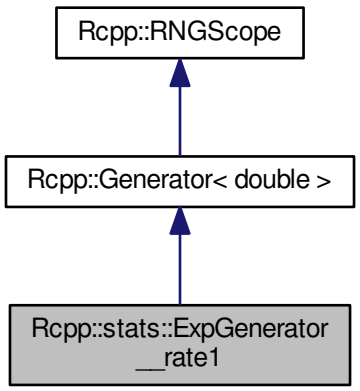
6.209 Rcpp::stats::ExpGenerator__rate1 Class Reference

`#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

6.209.1 Detailed Description

Definition at line 41 of file rexp.h.

6.209.2 Constructor & Destructor Documentation

6.209.2.1 Rcpp::stats::ExpGenerator__rate1::ExpGenerator__rate1 ( ) [inline]

Definition at line 43 of file rexp.h.

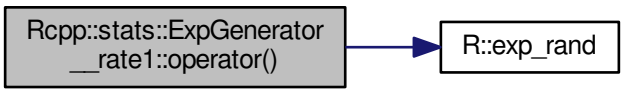
6.209.3 Member Function Documentation

6.209.3.1 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:

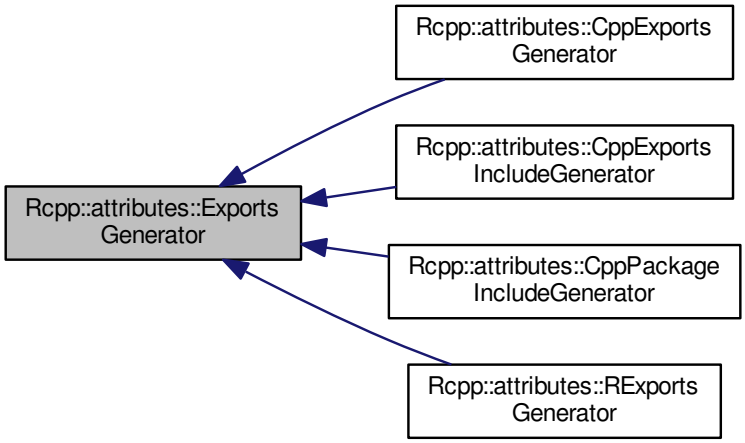


The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rexp.h

6.210 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
- virtual void writeBegin ()=0
- void writeFunctions (const SourceFileAttributes &attributes, bool verbose)
- virtual void writeEnd ()=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 hasCppInterface () const
- std::string exportValidationFunction ()
- std::string exportValidationFunctionRegisteredName ()
- std::string registerCCallableExportedName ()
- bool commit (const std::string &preamble=std::string())

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::ostringstream codeStream_
- bool hasCppInterface_

6.210.1 Detailed Description

Definition at line 551 of file attributes.cpp.

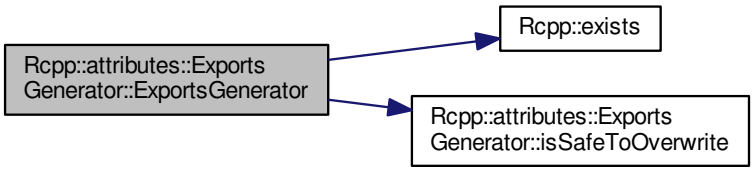
6.210.2 Constructor & Destructor Documentation

6.210.2.1 Rcpp::attributes::ExportsGenerator::ExportsGenerator (const std::string & *targetFile,* const std::string & *package,* const std::string & *commentPrefix*) [protected]

Definition at line 1724 of file attributes.cpp.

References existingCode_, Rcpp::exists(), isSafeToOverwrite(), packageCpp_, and targetFile_.

Here is the call graph for this function:



6.210.2.2 Rcpp::attributes::ExportsGenerator::ExportsGenerator (const ExportsGenerator &) [private]

6.210.2.3 virtual Rcpp::attributes::ExportsGenerator::~ExportsGenerator ( ) [inline], [virtual]

Definition at line 563 of file attributes.cpp.

6.210.3 Member Function Documentation

6.210.3.1 virtual bool Rcpp::attributes::ExportsGenerator::commit (const std::vector< std::string > & *includes*) [pure virtual]

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncludeGenerator, and Rcpp::attributes::CppExportsGenerator.

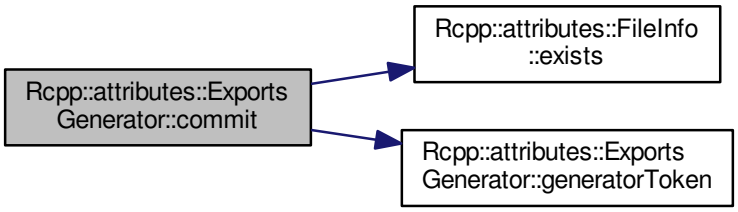
Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppPackageIncludeGenerator::commit(), and Rcpp::attributes::RExportsGenerator::commit().

6.210.3.2 bool Rcpp::attributes::ExportsGenerator::commit (const std::string & *preamble* = `std::string()`) [protected]

Definition at line 1763 of file attributes.cpp.

References codeStream_, commentPrefix_, existingCode_, Rcpp::attributes::FileInfo::exists(), generatorToken(), and targetFile_.

Here is the call graph for this function:



6.210.3.3 virtual void Rcpp::attributes::ExportsGenerator::doWriteFunctions (const SourceFileAttributes & *attributes*, bool *verbose*) [private], [pure virtual]

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncludeGenerator, and Rcpp::attributes::CppExportsGenerator.

Referenced by writeFunctions().

6.210.3.4 std::string Rcpp::attributes::ExportsGenerator::exportValidationFunction ( ) [inline], [protected]

Definition at line 598 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::writeEnd().

6.210.3.5 std::string Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName ( ) [inline], [protected]

Definition at line 601 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), and Rcpp::attributes::CppExports↩ Generator::writeEnd().

6.210.3.6 std::string Rcpp::attributes::ExportsGenerator::generatorToken ( ) const [inline], [private]

Definition at line 629 of file attributes.cpp.

Referenced by commit().

6.210.3.7 bool Rcpp::attributes::ExportsGenerator::hasCppInterface ( ) const [inline], [protected]

Definition at line 593 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator↩ ::commit(), Rcpp::attributes::CppPackageIncludeGenerator::commit(), Rcpp::attributes::CppExportsGenerator::write↩ End(), Rcpp::attributes::CppPackageIncludeGenerator::writeEnd(), and Rcpp::attributes::RExportsGenerator::write↩ End().

6.210.3.8 bool Rcpp::attributes::ExportsGenerator::isSafeToOverwrite ( ) const [inline], [private]

Definition at line 622 of file attributes.cpp.

Referenced by ExportsGenerator().

6.210.3.9 Rcpp::attributes::ExportsGenerator::operator std::ostream & ( ) [inline]

Definition at line 582 of file attributes.cpp.

6.210.3.10 ExportsGenerator& Rcpp::attributes::ExportsGenerator::operator= (const ExportsGenerator &) [private]

6.210.3.11 std::ostream& Rcpp::attributes::ExportsGenerator::ostr ( ) [inline], [protected]

Definition at line 589 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::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::RExportsGenerator::writeEnd().

6.210.3.12 const std::string& Rcpp::attributes::ExportsGenerator::package ( ) const [inline]

Definition at line 567 of file attributes.cpp.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::getCCallable(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), and Rcpp::attributes::RExportsGenerator::writeEnd().

6.210.3.13 const std::string& Rcpp::attributes::ExportsGenerator::packageCpp ( ) const [inline]

Definition at line 568 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::getHeaderGuard(), Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), and Rcpp::attributes::CppPackageIncludeGenerator::writeEnd().

6.210.3.14 std::string Rcpp::attributes::ExportsGenerator::registerCCallableExportedName ( ) [inline], [protected]

Definition at line 604 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::writeEnd(), and Rcpp::attributes::RExportsGenerator::writeEnd().

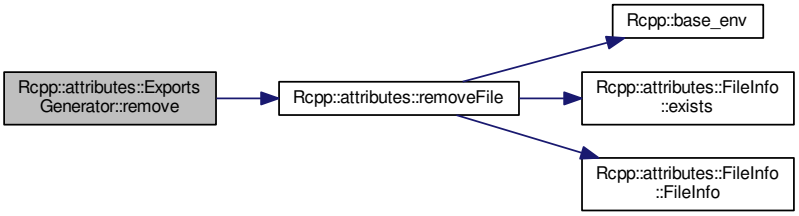
6.210.3.15 bool Rcpp::attributes::ExportsGenerator::remove ( )

Definition at line 1803 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:



6.210.3.16 const std::string& Rcpp::attributes::ExportsGenerator::targetFile ( ) const [inline]

Definition at line 566 of file attributes.cpp.

6.210.3.17 virtual void Rcpp::attributes::ExportsGenerator::writeBegin ( ) [pure virtual]

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncludeGenerator, and Rcpp::attributes::CppExportsGenerator.

6.210.3.18 virtual void Rcpp::attributes::ExportsGenerator::writeEnd ( ) [pure virtual]

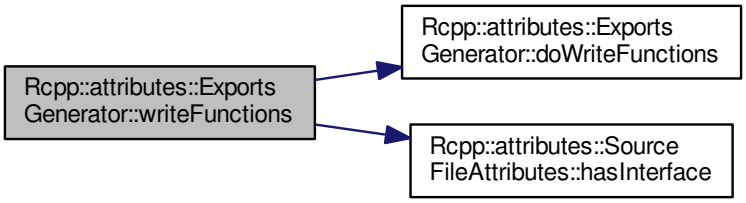
Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncludeGenerator, and Rcpp::attributes::CppExportsGenerator.

6.210.3.19 void Rcpp::attributes::ExportsGenerator::writeFunctions (const SourceFileAttributes & *attributes,* bool *verbose*)

Definition at line 1750 of file attributes.cpp.

References doWriteFunctions(), hasCppInterface_, Rcpp::attributes::SourceFileAttributes::hasInterface(), and Rcpp↩ ::attributes::kInterfaceCpp.

Here is the call graph for this function:



6.210.4 Member Data Documentation

6.210.4.1 std::ostringstream Rcpp::attributes::ExportsGenerator::codeStream_ [private]

Definition at line 639 of file attributes.cpp.

Referenced by commit().

6.210.4.2 std::string Rcpp::attributes::ExportsGenerator::commentPrefix_ [private]

Definition at line 637 of file attributes.cpp.

Referenced by commit().

6.210.4.3 std::string Rcpp::attributes::ExportsGenerator::existingCode_ [private]

Definition at line 638 of file attributes.cpp.

Referenced by commit(), and ExportsGenerator().

6.210.4.4 bool Rcpp::attributes::ExportsGenerator::hasCppInterface_ [private]

Definition at line 640 of file attributes.cpp.

Referenced by writeFunctions().

6.210.4.5 std::string Rcpp::attributes::ExportsGenerator::package_ [private]

Definition at line 635 of file attributes.cpp.

6.210.4.6 std::string Rcpp::attributes::ExportsGenerator::packageCpp_ [private]

Definition at line 636 of file attributes.cpp.

Referenced by ExportsGenerator().

6.210.4.7 std::string Rcpp::attributes::ExportsGenerator::targetFile_ [private]

Definition at line 634 of file attributes.cpp.

Referenced by commit(), ExportsGenerator(), and remove().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.211 Rcpp::attributes::ExportsGenerators Class Reference

Public Types

- typedef std::vector< ExportsGenerator * >::iterator Itr

Public Member Functions

- ExportsGenerators ()
- virtual ~ExportsGenerators ()
- void add (ExportsGenerator *pGenerator)
- void writeBegin ()
- void writeFunctions (const SourceFileAttributes &attributes, bool verbose)
- void writeEnd ()
- 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_

6.211.1 Detailed Description

Definition at line 726 of file attributes.cpp.

6.211.2 Member Typedef Documentation

6.211.2.1 typedef std::vector<ExportsGenerator*>::iterator Rcpp::attributes::ExportsGenerators::Itr

Definition at line 728 of file attributes.cpp.

6.211.3 Constructor & Destructor Documentation

6.211.3.1 Rcpp::attributes::ExportsGenerators::ExportsGenerators ( ) [inline]

Definition at line 730 of file attributes.cpp.

6.211.3.2 Rcpp::attributes::ExportsGenerators::~ExportsGenerators ( ) [virtual]

Definition at line 2289 of file attributes.cpp.

6.211.3.3 Rcpp::attributes::ExportsGenerators::ExportsGenerators (const ExportsGenerators &) [private]

6.211.4 Member Function Documentation

6.211.4.1 void Rcpp::attributes::ExportsGenerators::add (ExportsGenerator * *pGenerator*)

Definition at line 2298 of file attributes.cpp.

Referenced by compileAttributes().

6.211.4.2 std::vector< std::string > Rcpp::attributes::ExportsGenerators::commit (const std::vector< std::string > & *includes*)

Definition at line 2320 of file attributes.cpp.

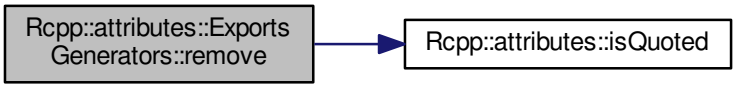
6.211.4.3 ExportsGenerators& Rcpp::attributes::ExportsGenerators::operator= (const ExportsGenerators &) [private]

6.211.4.4 std::vector< std::string > Rcpp::attributes::ExportsGenerators::remove ( )

Definition at line 2334 of file attributes.cpp.

References Rcpp::attributes::isQuoted().

Here is the call graph for this function:



6.211.4.5 void Rcpp::attributes::ExportsGenerators::writeBegin ( )

Definition at line 2302 of file attributes.cpp.

6.211.4.6 void Rcpp::attributes::ExportsGenerators::writeEnd ( )

Definition at line 2314 of file attributes.cpp.

6.211.4.7 void Rcpp::attributes::ExportsGenerators::writeFunctions (const SourceFileAttributes & *attributes*, bool *verbose*)

Definition at line 2307 of file attributes.cpp.

6.211.5 Member Data Documentation

6.211.5.1 std::vector<ExportsGenerator*> Rcpp::attributes::ExportsGenerators::generators_ [private]

Definition at line 753 of file attributes.cpp.

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.212 Rcpp::traits::Extractor< RTYPE, NA, VECTOR > Struct Template Reference

`#include <Extractor.h>`

Public Types

- typedef VECTOR type

6.212.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
struct Rcpp::traits::Extractor< RTYPE, NA, VECTOR >

Definition at line 29 of file Extractor.h.

6.212.2 Member Typedef Documentation

6.212.2.1 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

6.213 tinyformat::detail::is_convertible< T1, T2 >::fail Struct Reference

Public Attributes

- char dummy [2]

6.213.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::fail

Definition at line 189 of file tinyformat.h.

6.213.2 Member Data Documentation

6.213.2.1 template<typename T1 , typename T2 > char tinyformat::detail::is_convertible< T1, T2 >::fail::dummy[2]

Definition at line 189 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.214 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

6.214.1 Detailed Description

template<typename VECTOR>
class Rcpp::Fast< VECTOR >

Definition at line 27 of file Fast.h.

6.214.2 Member Typedef Documentation

6.214.2.1 template<typename VECTOR> typedef VECTOR::stored_type Rcpp::Fast< VECTOR >::value_type

Definition at line 29 of file Fast.h.

6.214.3 Constructor & Destructor Documentation

6.214.3.1 template<typename VECTOR> Rcpp::Fast< VECTOR >::Fast (VECTOR & *v_*) [inline]

Definition at line 31 of file Fast.h.

6.214.4 Member Function Documentation

6.214.4.1 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.

6.214.4.2 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.

6.214.4.3 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.

6.214.5 Member Data Documentation

6.214.5.1 template<typename VECTOR> value_type* Rcpp::Fast< VECTOR >::data [private]

Definition at line 39 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::operator[ ]().

6.214.5.2 template<typename VECTOR> VECTOR& Rcpp::Fast< VECTOR >::v [private]

Definition at line 38 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::size().

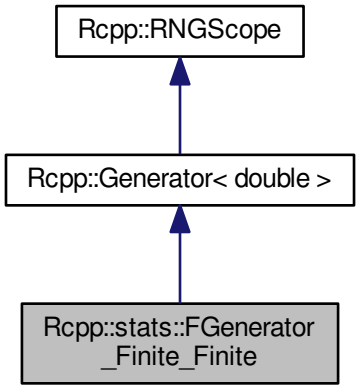
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Fast.h

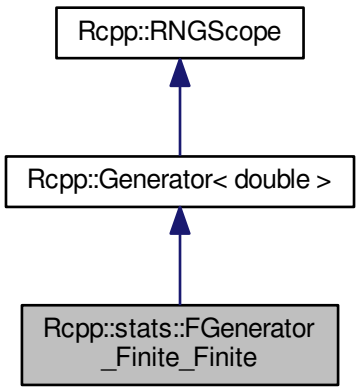
6.215 Rcpp::stats::FGenerator_Finite_Finite Class Reference

`#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

6.215.1 Detailed Description

Definition at line 28 of file rf.h.

6.215.2 Constructor & Destructor Documentation

6.215.2.1 Rcpp::stats::FGenerator_Finite_Finite::FGenerator_Finite_Finite (double *n1_*, double *n2_*) [inline]

Definition at line 31 of file rf.h.

6.215.3 Member Function Documentation

6.215.3.1 double Rcpp::stats::FGenerator_Finite_Finite::operator() ( ) const [inline]

Definition at line 34 of file rf.h.

References n1__2, n2__2, and ratio.

6.215.4 Member Data Documentation

6.215.4.1 double Rcpp::stats::FGenerator_Finite_Finite::n1__2 [private]

Definition at line 41 of file rf.h.

Referenced by operator()(), and Rcpp::stats::FGenerator_Finite_NotFinite::operator()().

6.215.4.2 double Rcpp::stats::FGenerator_Finite_Finite::n2__2 [private]

Definition at line 41 of file rf.h.

Referenced by operator()(), and Rcpp::stats::FGenerator_NotFinite_Finite::operator()().

6.215.4.3 double Rcpp::stats::FGenerator_Finite_Finite::ratio [private]

Definition at line 41 of file rf.h.

Referenced by operator()().

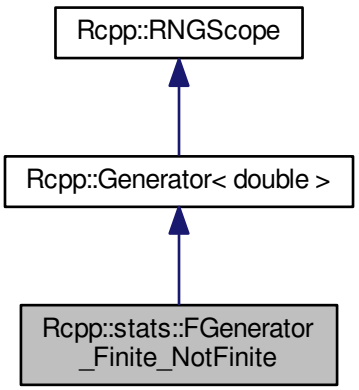
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

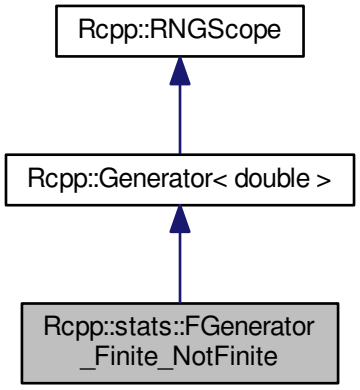
6.216 Rcpp::stats::FGenerator_Finite_NotFinite Class Reference

```
#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

6.216.1 Detailed Description

Definition at line 58 of file rf.h.

6.216.2 Constructor & Destructor Documentation

6.216.2.1 Rcpp::stats::FGenerator_Finite_NotFinite::FGenerator_Finite_NotFinite (double *n1_*) [inline]

Definition at line 61 of file rf.h.

6.216.3 Member Function Documentation

6.216.3.1 double Rcpp::stats::FGenerator_Finite_NotFinite::operator() ( ) const [inline]

Definition at line 63 of file rf.h.

References Rcpp::stats::FGenerator_Finite_Finite::n1__2.

6.216.4 Member Data Documentation

6.216.4.1 double Rcpp::stats::FGenerator_Finite_NotFinite::n1 [private]

Definition at line 69 of file rf.h.

6.216.4.2 double Rcpp::stats::FGenerator_Finite_NotFinite::n1__2 [private]

Definition at line 69 of file rf.h.

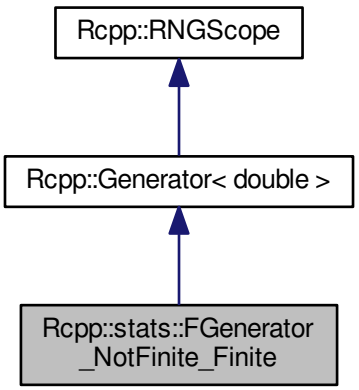
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

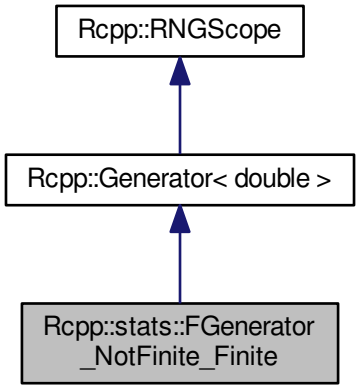
6.217 Rcpp::stats::FGenerator_NotFinite_Finite Class Reference

`#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

6.217.1 Detailed Description

Definition at line 44 of file rf.h.

6.217.2 Constructor & Destructor Documentation

6.217.2.1 Rcpp::stats::FGenerator_NotFinite_Finite::FGenerator_NotFinite_Finite (double *n2_*) [inline]

Definition at line 47 of file rf.h.

6.217.3 Member Function Documentation

6.217.3.1 double Rcpp::stats::FGenerator_NotFinite_Finite::operator() ( ) const [inline]

Definition at line 49 of file rf.h.

References Rcpp::stats::FGenerator_Finite_Finite::n2__2.

6.217.4 Member Data Documentation

6.217.4.1 double Rcpp::stats::FGenerator_NotFinite_Finite::n2 [private]

Definition at line 55 of file rf.h.

6.217.4.2 double Rcpp::stats::FGenerator_NotFinite_Finite::n2__2 [private]

Definition at line 55 of file rf.h.

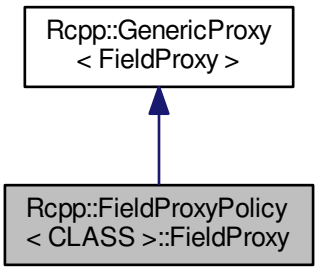
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

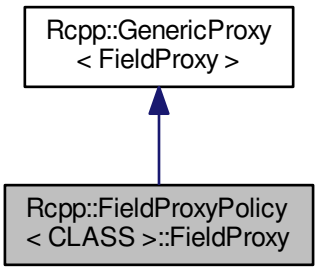
6.218 Rcpp::FieldProxyPolicy< CLASS >::FieldProxy Class Reference

`#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

6.218.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >::FieldProxy

Definition at line 27 of file FieldProxy.h.

6.218.2 Constructor & Destructor Documentation

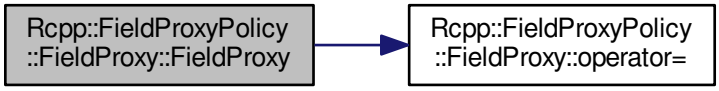
6.218.2.1 template<typename CLASS > Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::FieldProxy (CLASS & *v*, const std::string & *name*) [inline]

Definition at line 29 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=().

Referenced by Rcpp::FieldProxyPolicy< CLASS >::field().

Here is the call graph for this function:



6.218.3 Member Function Documentation

6.218.3.1 template<typename CLASS > SEXP Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get ( ) const [inline], [private]

Definition at line 43 of file FieldProxy.h.

References Rcpp::Rcpp_eval().

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=().

Here is the call graph for this function:

6.218.3.2 template<typename CLASS > Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator SEXP ( ) const [inline]

Definition at line 37 of file FieldProxy.h.

6.218.3.3 template<typename CLASS > template<typename T > Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator T ( ) const

Definition at line 195 of file proxy.h.

6.218.3.4 template<typename CLASS > FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (const FieldProxy & *rhs*)

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::FieldProxy().

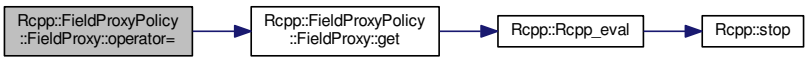
6.218.3.5 template<typename CLASS > template<typename T > FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (const T & *rhs*)

6.218.3.6 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:

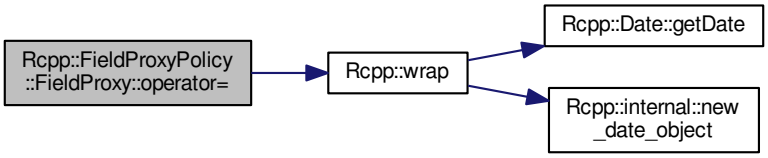


6.218.3.7 template<typename CLASS > template<typename T > FieldProxyPolicy<CLASS>::FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (const T & *rhs*)

Definition at line 186 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.218.3.8 template<typename CLASS > void Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set (SEXP *x*) [inline], [private]

Definition at line 47 of file FieldProxy.h.

References Rcpp::Rcpp_eval().

Here is the call graph for this function:

6.218.4 Member Data Documentation

6.218.4.1 template<typename CLASS > const std::string& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field_name [private]

Definition at line 41 of file FieldProxy.h.

6.218.4.2 template<typename CLASS > CLASS& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent [private]

Definition at line 40 of file FieldProxy.h.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/FieldProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.219 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

6.219.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >

Definition at line 24 of file FieldProxy.h.

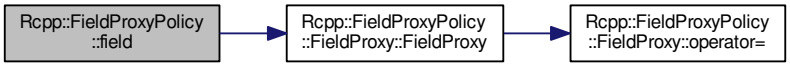
6.219.2 Member Function Documentation

6.219.2.1 template<typename CLASS > FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (const std::string & *name*) [inline]

Definition at line 74 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::FieldProxy().

Here is the call graph for this function:



6.219.2.2 template<typename CLASS > const_FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (const std::string & *name*) const [inline]

Definition at line 77 of file FieldProxy.h.

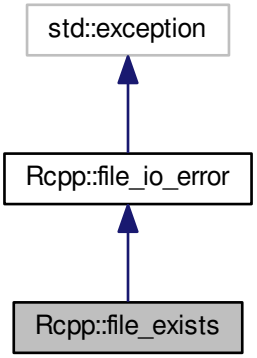
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/FieldProxy.h

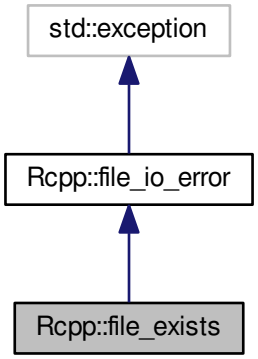
6.220 Rcpp::file_exists Class Reference

```
#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 ()

6.220.1 Detailed Description

Definition at line 78 of file exceptions.h.

6.220.2 Constructor & Destructor Documentation

6.220.2.1 Rcpp::file_exists::file_exists (const std::string & *file*) throw) [inline]

Definition at line 80 of file exceptions.h.

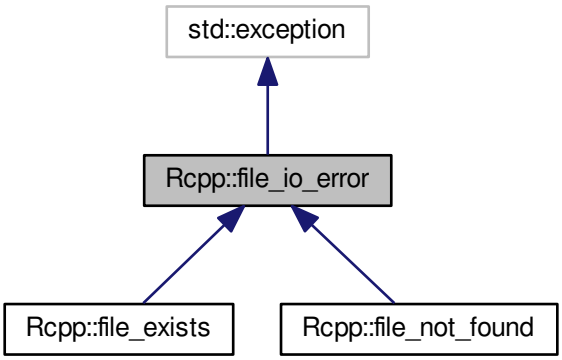
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

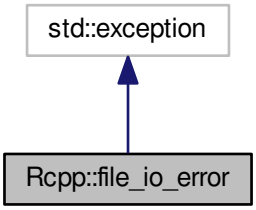
6.221 Rcpp::file_io_error Class Reference

`#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

6.221.1 Detailed Description

Definition at line 60 of file exceptions.h.

6.221.2 Constructor & Destructor Documentation

6.221.2.1 Rcpp::file_io_error::file_io_error (const std::string & *file*) throw) [inline]

Definition at line 62 of file exceptions.h.

6.221.2.2 Rcpp::file_io_error::file_io_error (int *code,* const std::string & *file*) throw) [inline]

Definition at line 63 of file exceptions.h.

6.221.2.3 Rcpp::file_io_error::file_io_error (const std::string & *msg,* const std::string & *file*) throw) [inline]

Definition at line 64 of file exceptions.h.

6.221.2.4 virtual Rcpp::file_io_error::~file_io_error ( ) throw) [inline], [virtual]

Definition at line 65 of file exceptions.h.

6.221.3 Member Function Documentation

6.221.3.1 std::string Rcpp::file_io_error::filePath ( ) const throw) [inline]

Definition at line 67 of file exceptions.h.

Referenced by compileAttributes().

6.221.3.2 virtual const char* Rcpp::file_io_error::what ( ) const throw) [inline], [virtual]

Definition at line 66 of file exceptions.h.

References Rcpp::exception::message.

6.221.4 Member Data Documentation

6.221.4.1 std::string Rcpp::file_io_error::file [private]

Definition at line 70 of file exceptions.h.

6.221.4.2 std::string Rcpp::file_io_error::message [private]

Definition at line 67 of file exceptions.h.

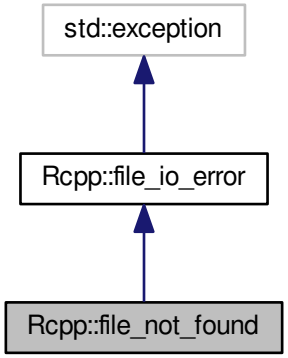
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

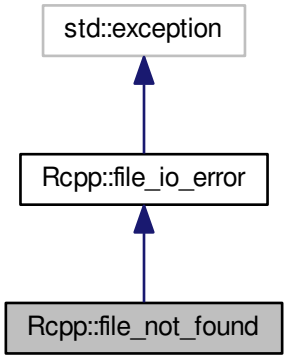
6.222 Rcpp::file_not_found Class Reference

```
#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 ()

6.222.1 Detailed Description

Definition at line 73 of file exceptions.h.

6.222.2 Constructor & Destructor Documentation

6.222.2.1 Rcpp::file_not_found::file_not_found (const std::string & *file*) throw) [inline]

Definition at line 75 of file exceptions.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

6.223 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_

6.223.1 Detailed Description

Definition at line 49 of file attributes.cpp.

6.223.2 Constructor & Destructor Documentation

6.223.2.1 Rcpp::attributes::FileInfo::FileInfo (const std::string & *path*) [explicit]

Definition at line 2673 of file attributes.cpp.

References exists_, and lastModified_.

Referenced by Rcpp::attributes::createDirectory(), Rcpp::attributes::isRoxygenCpp(), and Rcpp::attributes::removeFile().

6.223.2.2 Rcpp::attributes::FileInfo::FileInfo (const List & *fileInfo*) [inline], [explicit]

Definition at line 56 of file attributes.cpp.

References exists_, lastModified_, and path_.

6.223.3 Member Function Documentation

6.223.3.1 bool Rcpp::attributes::FileInfo::exists ( ) const [inline]

Definition at line 72 of file attributes.cpp.

References exists_.

Referenced by Rcpp::attributes::ExportsGenerator::commit(), Rcpp::attributes::createDirectory(), Rcpp::attributes::isRoxygenCpp(), and Rcpp::attributes::removeFile().

6.223.3.2 std::string Rcpp::attributes::FileInfo::extension ( ) const [inline]

Definition at line 75 of file attributes.cpp.

References path_.

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

6.223.3.3 double Rcpp::attributes::FileInfo::lastModified ( ) const [inline]

Definition at line 73 of file attributes.cpp.

References lastModified_.

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

6.223.3.4 bool Rcpp::attributes::FileInfo::operator!= (const FileInfo & *other*) const [inline]

Definition at line 93 of file attributes.cpp.

6.223.3.5 bool Rcpp::attributes::FileInfo::operator< (const FileInfo & *other*) const [inline]

Definition at line 83 of file attributes.cpp.

References path_.

6.223.3.6 std::ostream& Rcpp::attributes::FileInfo::operator<< (std::ostream & *os*) const [inline]

Definition at line 97 of file attributes.cpp.

References path_.

6.223.3.7 bool Rcpp::attributes::FileInfo::operator== (const FileInfo & *other*) const [inline]

Definition at line 87 of file attributes.cpp.

References exists_, lastModified_, and path_.

6.223.3.8 std::string Rcpp::attributes::FileInfo::path ( ) const [inline]

Definition at line 71 of file attributes.cpp.

References path_.

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

6.223.3.9 List Rcpp::attributes::FileInfo::toList ( ) const [inline]

Definition at line 63 of file attributes.cpp.

References exists_, lastModified_, and path_.

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

6.223.4 Member Data Documentation

6.223.4.1 bool Rcpp::attributes::FileInfo::exists_ [private]

Definition at line 104 of file attributes.cpp.

Referenced by exists(), FileInfo(), operator==(), and toList().

6.223.4.2 double Rcpp::attributes::FileInfo::lastModified_ [private]

Definition at line 105 of file attributes.cpp.

Referenced by FileInfo(), lastModified(), operator==(), and toList().

6.223.4.3 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

6.224 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

6.224.1 Detailed Description

template<typename RESULT_TYPE = SEXP>
class Rcpp::fixed_call< RESULT_TYPE >

Definition at line 164 of file Language.h.

6.224.2 Member Typedef Documentation

6.224.2.1 template<typename RESULT_TYPE = SEXP> typedef RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::result_type

Definition at line 166 of file Language.h.

6.224.3 Constructor & Destructor Documentation

6.224.3.1 template<typename RESULT_TYPE = SEXP> Rcpp::fixed_call< RESULT_TYPE >::fixed_call (Language *call_*) [inline]

Definition at line 168 of file Language.h.

6.224.3.2 template<typename RESULT_TYPE = SEXP> Rcpp::fixed_call< RESULT_TYPE >::fixed_call (Function *fun*) [inline]

Definition at line 169 of file Language.h.

6.224.4 Member Function Documentation

6.224.4.1 template<typename RESULT_TYPE = SEXP> RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::operator() ( ) [inline]

Definition at line 171 of file Language.h.

References Rcpp::fixed_call< RESULT_TYPE >::call.

6.224.5 Member Data Documentation

6.224.5.1 template<typename RESULT_TYPE = SEXP> Language Rcpp::fixed_call< RESULT_TYPE >::call [private]

Definition at line 176 of file Language.h.

Referenced by Rcpp::fixed_call< RESULT_TYPE >::operator()(), Rcpp::unary_call< T, RESULT_TYPE >::operator()(), and Rcpp::binary_call< T1, T2, RESULT_TYPE >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.225 Rcpp::sugar::forbidden_conversion< bool > Class Template Reference

`#include <SingleLogicalResult.h>`

6.225.1 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

6.226 Rcpp::sugar::forbidden_conversion< true > Class Template Reference

`#include <SingleLogicalResult.h>`

6.226.1 Detailed Description

template<>
class Rcpp::sugar::forbidden_conversion< true >

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

6.227 tinyformat::detail::FormatArg Class Reference

`#include <tinyformat.h>`

Public Member Functions

- FormatArg ()
- template<typename T >
  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

- template<typename T >
  static TINYFORMAT_HIDDEN void formatImpl (std::ostream &out, const char *fmtBegin, const char *fmtEnd, int ntrunc, const void *value)
- template<typename T >
  static TINYFORMAT_HIDDEN int 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)

6.227.1 Detailed Description

Definition at line 501 of file tinyformat.h.

6.227.2 Constructor & Destructor Documentation

6.227.2.1 tinyformat::detail::FormatArg::FormatArg ( ) [inline]

Definition at line 504 of file tinyformat.h.

6.227.2.2 template<typename T > tinyformat::detail::FormatArg::FormatArg (const T & *value*) [inline]

Definition at line 507 of file tinyformat.h.

6.227.3 Member Function Documentation

6.227.3.1 void tinyformat::detail::FormatArg::format (std::ostream & *out,* const char * *fmtBegin,* const char * *fmtEnd,* int *ntrunc*) const [inline]

Definition at line 513 of file tinyformat.h.

Referenced by tinyformat::detail::formatImpl().

6.227.3.2 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 526 of file tinyformat.h.

References tinyformat::formatValue().

Here is the call graph for this function:

6.227.3.3 int tinyformat::detail::FormatArg::toInt ( ) const [inline]

Definition at line 519 of file tinyformat.h.

Referenced by tinyformat::detail::streamStateFromFormat().

6.227.3.4 template<typename T > static TINYFORMAT_HIDDEN int tinyformat::detail::FormatArg::toIntImpl (const void * *value*) [inline], [static], [private]

Definition at line 533 of file tinyformat.h.

6.227.4 Member Data Documentation

6.227.4.1 void(* tinyformat::detail::FormatArg::m_formatImpl) (std::ostream &out, const char *fmtBegin, const char *fmtEnd, int ntrunc, const void *value) [private]

Definition at line 539 of file tinyformat.h.

6.227.4.2 int(* tinyformat::detail::FormatArg::m_toIntImpl) (const void *value) [private]

Definition at line 541 of file tinyformat.h.

6.227.4.3 const void* tinyformat::detail::FormatArg::m_value [private]

Definition at line 538 of file tinyformat.h.

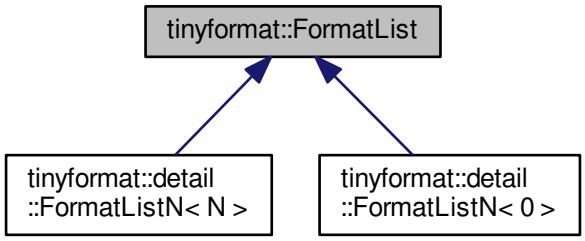
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

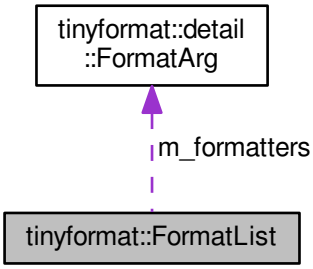
6.228 tinyformat::FormatList Class Reference

`#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)

6.228.1 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 843 of file tinyformat.h.

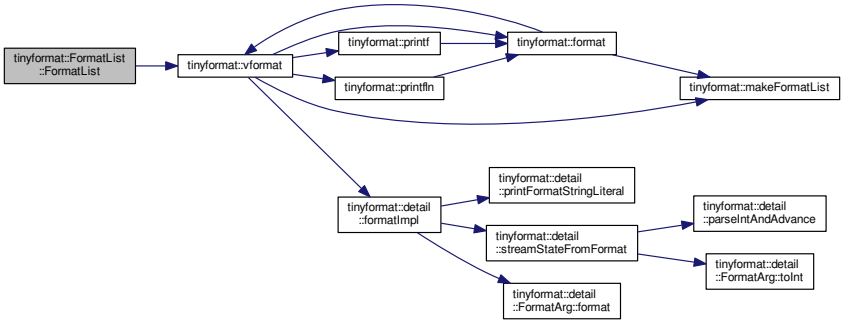
6.228.2 Constructor & Destructor Documentation

6.228.2.1 tinyformat::FormatList::FormatList (detail::FormatArg * *formatters,* int *N*) [inline]

Definition at line 846 of file tinyformat.h.

References tinyformat::vformat().

Here is the call graph for this function:



6.228.3 Friends And Related Function Documentation

6.228.3.1 void vformat (std::ostream & *out,* const char * *fmt,* const FormatList & *list*) [friend]

6.228.4 Member Data Documentation

6.228.4.1 const detail::FormatArg* tinyformat::FormatList::m_formatters [private]

Definition at line 853 of file tinyformat.h.

Referenced by tinyformat::vformat().

6.228.4.2 int tinyformat::FormatList::m_N [private]

Definition at line 854 of file tinyformat.h.

Referenced by tinyformat::vformat().

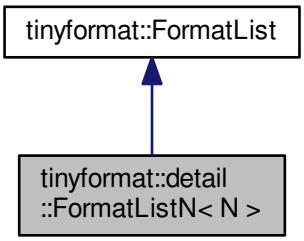
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

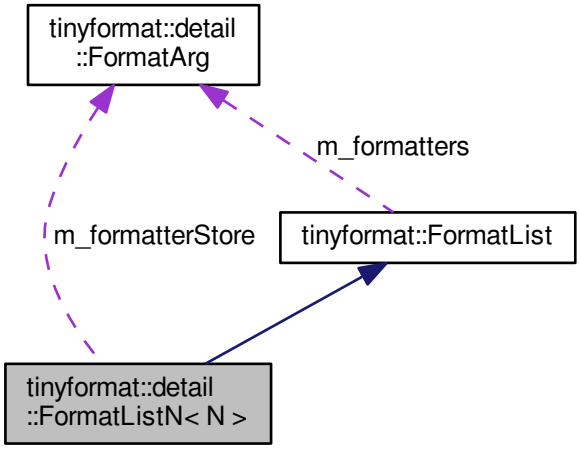
6.229 tinyformat::detail::FormatListN< N > Class Template Reference

`#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]

6.229.1 Detailed Description

template<int N>
class tinyformat::detail::FormatListN< N >

Definition at line 865 of file tinyformat.h.

6.229.2 Member Function Documentation

6.229.2.1 template<int N> void tinyformat::detail::FormatListN< N >::init (int) [inline]

Definition at line 875 of file tinyformat.h.

6.229.3 Member Data Documentation

6.229.3.1 template<int N> FormatArg tinyformat::detail::FormatListN< N >::m_formatterStore[N] [private]

Definition at line 898 of file tinyformat.h.

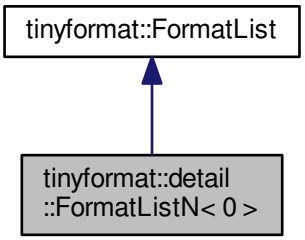
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

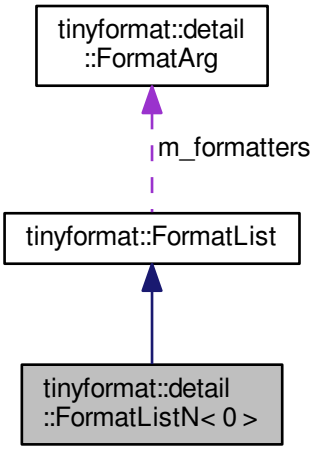
6.230 tinyformat::detail::FormatListN< 0 > Class Template Reference

`#include <tinyformat.h>`

Inheritance diagram for tinyformat::detail::FormatListN< 0 >:



Collaboration diagram for tinyformat::detail::FormatListN< 0 >:



Public Member Functions

- FormatListN ()

6.230.1 Detailed Description

template<>
class tinyformat::detail::FormatListN< 0 >

Definition at line 903 of file tinyformat.h.

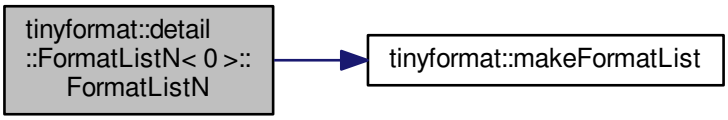
6.230.2 Constructor & Destructor Documentation

6.230.2.1 tinyformat::detail::FormatListN< 0 >::FormatListN ( ) [inline]

Definition at line 905 of file tinyformat.h.

References tinyformat::makeFormatList().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.231 tinyformat::detail::formatValueAsType< T, fmtT, convertible > Struct Template Reference

`#include <tinyformat.h>`

Static Public Member Functions

- static void invoke (std::ostream &, const T &)

6.231.1 Detailed Description

template<typename T, typename fmtT, bool convertible = is_convertible<T, fmtT>::value>
struct tinyformat::detail::formatValueAsType< T, fmtT, convertible >

Definition at line 225 of file tinyformat.h.

6.231.2 Member Function Documentation

6.231.2.1 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 227 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.232 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)

6.232.1 Detailed Description

template<typename T, typename fmtT>
struct tinyformat::detail::formatValueAsType< T, fmtT, true >

Definition at line 232 of file tinyformat.h.

6.232.2 Member Function Documentation

6.232.2.1 template<typename T , typename fmtT > static void tinyformat::detail::formatValueAsType< T, fmtT, true >::invoke (std::ostream & *out*, const T & *value*) [inline], [static]

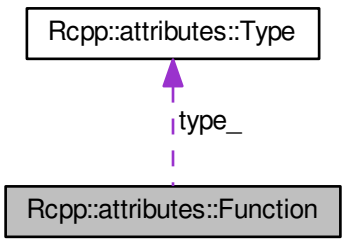
Definition at line 234 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.233 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_

6.233.1 Detailed Description

Definition at line 237 of file attributes.cpp.

6.233.2 Constructor & Destructor Documentation

6.233.2.1 Rcpp::attributes::Function::Function ( ) [inline]

Definition at line 239 of file attributes.cpp.

6.233.2.2 Rcpp::attributes::Function::Function (const Type & *type*, const std::string & *name*, const std::vector< Argument > & *arguments*) [inline]

Definition at line 240 of file attributes.cpp.

6.233.3 Member Function Documentation

6.233.3.1 const std::vector<Argument>& Rcpp::attributes::Function::arguments ( ) const [inline]

Definition at line 272 of file attributes.cpp.

6.233.3.2 bool Rcpp::attributes::Function::empty ( ) const [inline]

Definition at line 258 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.233.3.3 bool Rcpp::attributes::Function::isHidden ( ) const [inline]

Definition at line 254 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

6.233.3.4 const std::string& Rcpp::attributes::Function::name ( ) const [inline]

Definition at line 271 of file attributes.cpp.

Referenced by Rcpp::attributes::generateCpp(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

6.233.3.5 bool Rcpp::attributes::Function::operator!= (const Function & *other*) const [inline]

Definition at line 266 of file attributes.cpp.

6.233.3.6 bool Rcpp::attributes::Function::operator== (const Function & *other*) const [inline]

Definition at line 260 of file attributes.cpp.

References arguments_, name_, and type_.

6.233.3.7 Function Rcpp::attributes::Function::renamedTo (const std::string & *name*) const [inline]

Definition at line 247 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), and Rcpp::attributes::CppExportsInclude↩
Generator::doWriteFunctions().

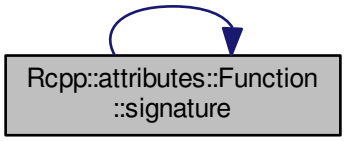
6.233.3.8 std::string Rcpp::attributes::Function::signature ( ) const [inline]

Definition at line 251 of file attributes.cpp.

References signature().

Referenced by signature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



6.233.3.9 std::string Rcpp::attributes::Function::signature (const std::string & *name*) const

Definition at line 974 of file attributes.cpp.

6.233.3.10 const Type& Rcpp::attributes::Function::type ( ) const [inline]

Definition at line 270 of file attributes.cpp.

6.233.4 Member Data Documentation

6.233.4.1 std::vector<Argument> Rcpp::attributes::Function::arguments_ [private]

Definition at line 277 of file attributes.cpp.

Referenced by operator==().

6.233.4.2 std::string Rcpp::attributes::Function::name_ [private]

Definition at line 276 of file attributes.cpp.

Referenced by operator==().

6.233.4.3 Type Rcpp::attributes::Function::type_ [private]

Definition at line 275 of file attributes.cpp.

Referenced by operator==().

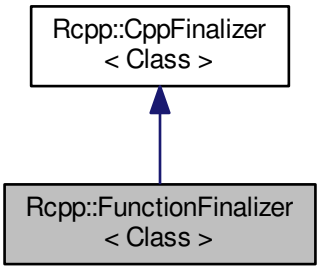
The documentation for this class was generated from the following file:

- src/attributes.cpp

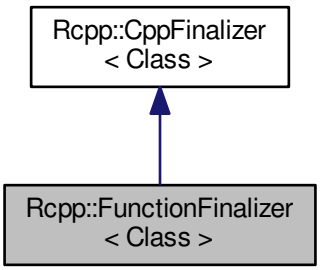
6.234 Rcpp::FunctionFinalizer< Class > Class Template Reference

`#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

6.234.1 Detailed Description

template<typename Class>
class Rcpp::FunctionFinalizer< Class >

Definition at line 307 of file Module.h.

6.234.2 Member Typedef Documentation

6.234.2.1 template<typename Class > typedef void(∗ Rcpp::FunctionFinalizer< Class >::Pointer) (Class ∗)

Definition at line 309 of file Module.h.

6.234.3 Constructor & Destructor Documentation

6.234.3.1 template<typename Class > Rcpp::FunctionFinalizer< Class >::FunctionFinalizer (Pointer *p*) [inline]

Definition at line 310 of file Module.h.

6.234.4 Member Function Documentation

6.234.4.1 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.

6.234.5 Member Data Documentation

6.234.5.1 template<typename Class > Pointer Rcpp::FunctionFinalizer< Class >::finalizer [private]

Definition at line 317 of file Module.h.

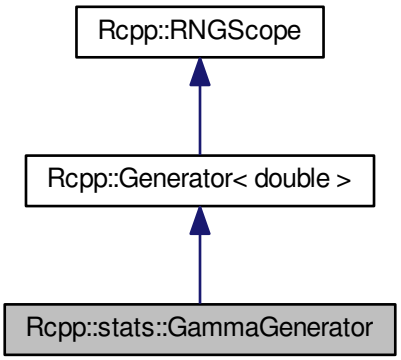
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

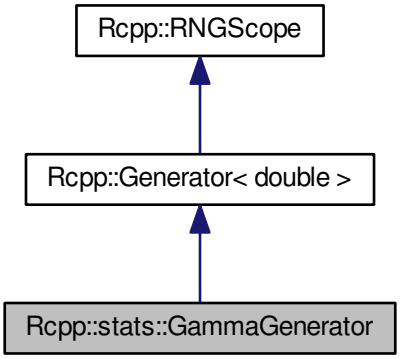
6.235 Rcpp::stats::GammaGenerator Class Reference

```
#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

6.235.1 Detailed Description

Definition at line 28 of file rgamma.h.

6.235.2 Constructor & Destructor Documentation

6.235.2.1 Rcpp::stats::GammaGenerator::GammaGenerator (double *a_*, double *scale_*) [inline]

Definition at line 30 of file rgamma.h.

6.235.3 Member Function Documentation

6.235.3.1 double Rcpp::stats::GammaGenerator::operator() ( ) const [inline]

Definition at line 32 of file rgamma.h.

References a, and scale.

6.235.4 Member Data Documentation

6.235.4.1 double Rcpp::stats::GammaGenerator::a [private]

Definition at line 34 of file rgamma.h.

Referenced by operator()().

6.235.4.2 double Rcpp::stats::GammaGenerator::scale [private]

Definition at line 34 of file rgamma.h.

Referenced by operator()().

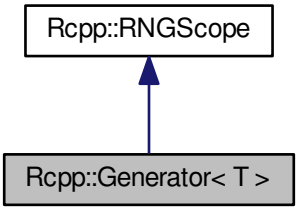
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rgamma.h

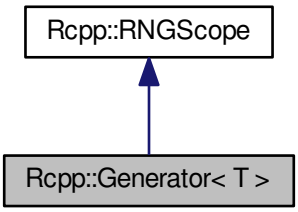
6.236 Rcpp::Generator< T > Class Template Reference

`#include <random.h>`

Inheritance diagram for Rcpp::Generator< T >:



Collaboration diagram for Rcpp::Generator< T >:



Public Types

- typedef T r_generator

Additional Inherited Members

6.236.1 Detailed Description

template<typename T>
class Rcpp::Generator< T >

Definition at line 34 of file random.h.

6.236.2 Member Typedef Documentation

6.236.2.1 template<typename T> typedef T Rcpp::Generator< T >::r_generator

Definition at line 36 of file random.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/random.h

6.237 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)

6.237.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::generic_element_converter< RTYPE >

Definition at line 74 of file converter.h.

6.237.2 Member Typedef Documentation

6.237.2.1 template<int RTYPE> typedef SEXP Rcpp::internal::generic_element_converter< RTYPE >::target

Definition at line 76 of file converter.h.

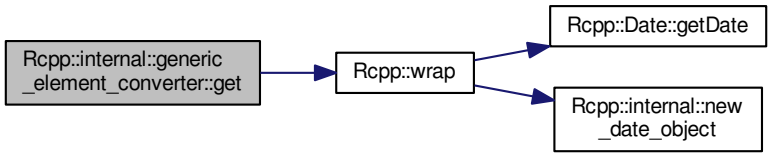
6.237.3 Member Function Documentation

6.237.3.1 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:

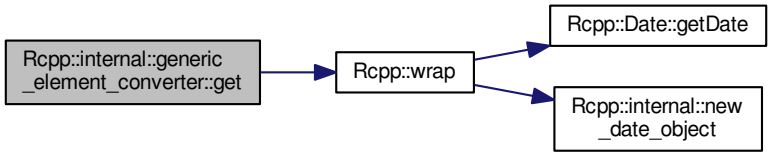


6.237.3.2 template<int RTYPE> static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (const char * *input*) [inline], [static]

Definition at line 83 of file converter.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.237.3.3 template<int RTYPE> static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (SEXP *input*) [inline], [static]

Definition at line 87 of file converter.h.

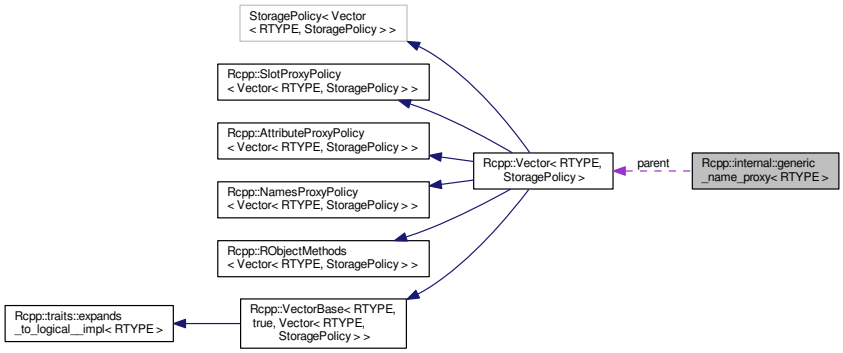
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.238 Rcpp::internal::generic_name_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Collaboration diagram for Rcpp::internal::generic_name_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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

6.238.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::generic_name_proxy< RTYPE >

Definition at line 34 of file 00_forward_proxy.h.

6.238.2 Member Typedef Documentation

6.238.2.1 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::generic_name_proxy< RTYPE >::VECTOR

Definition at line 136 of file proxy.h.

6.238.3 Constructor & Destructor Documentation

6.238.3.1 template<int RTYPE> Rcpp::internal::generic_name_proxy< RTYPE >::generic_name_proxy (VECTOR & *v*, const std::string & *name_*) [inline]

Definition at line 137 of file proxy.h.

6.238.3.2 template<int RTYPE> Rcpp::internal::generic_name_proxy< RTYPE >::generic_name_proxy (const generic_name_proxy< RTYPE > & *other*) [inline]

Definition at line 139 of file proxy.h.

6.238.3.3 template<int RTYPE> Rcpp::internal::generic_name_proxy< RTYPE >::~generic_name_proxy ( ) [inline]

Definition at line 141 of file proxy.h.

6.238.4 Member Function Documentation

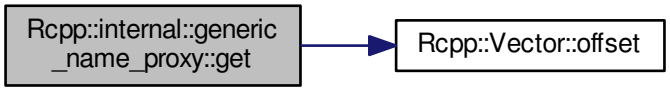
6.238.4.1 template<int RTYPE> SEXP Rcpp::internal::generic_name_proxy< RTYPE >::get ( ) const [inline], [private]

Definition at line 187 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Referenced by Rcpp::internal::generic_name_proxy< RTYPE >::operator=().

Here is the call graph for this function:



6.238.4.2 template<int RTYPE> Rcpp::internal::generic_name_proxy< RTYPE >::operator SEXP ( ) const [inline]

Definition at line 160 of file proxy.h.

6.238.4.3 template<int RTYPE> template<typename T > Rcpp::internal::generic_name_proxy< RTYPE >::operator T ( ) const [inline]

Definition at line 165 of file proxy.h.

References RCPP_DEBUG_1.

6.238.4.4 template<int RTYPE> generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE >::operator= (SEXP *rhs*) [inline]

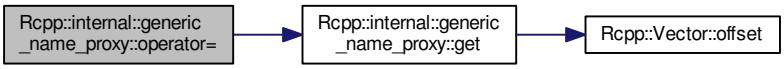
Definition at line 143 of file proxy.h.

6.238.4.5 template<int RTYPE> generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE >::operator= (const generic_name_proxy< RTYPE > & *other*) [inline]

Definition at line 147 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE >::get().

Here is the call graph for this function:

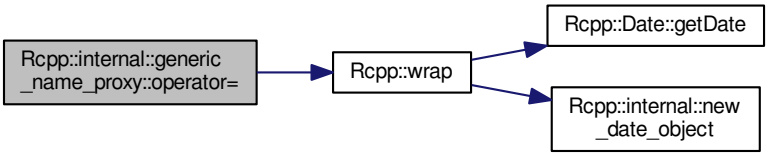


6.238.4.6 template<int RTYPE> template<typename T > generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE >::operator= (const T & *rhs*) [inline]

Definition at line 153 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:

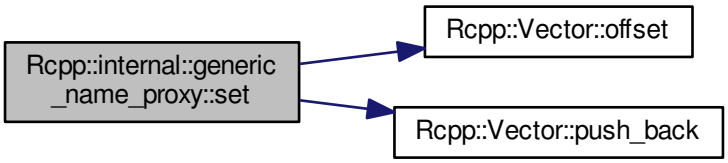


6.238.4.7 template<int RTYPE> void Rcpp::internal::generic_name_proxy< RTYPE >::set (SEXP *rhs*) [inline], [private]

Definition at line 178 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Here is the call graph for this function:



6.238.5 Member Data Documentation

6.238.5.1 template<int RTYPE> std::string Rcpp::internal::generic_name_proxy< RTYPE >::name [private]

Definition at line 177 of file proxy.h.

6.238.5.2 template<int RTYPE> VECTOR& Rcpp::internal::generic_name_proxy< RTYPE >::parent [private]

Definition at line 176 of file proxy.h.

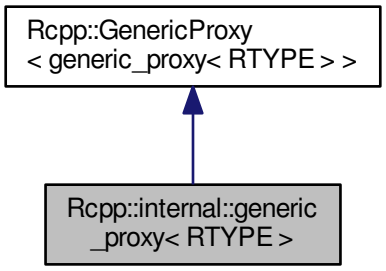
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

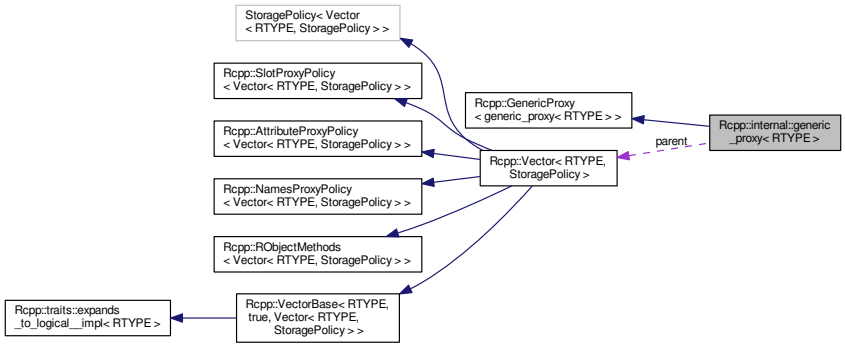
6.239 Rcpp::internal::generic_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Inheritance diagram for Rcpp::internal::generic_proxy< RTYPE >:



Collaboration diagram for Rcpp::internal::generic_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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<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)

Public Attributes

- VECTOR * parent
- R_xlen_t index

Private Member Functions

- void set (SEXP x)
- SEXP get () const

6.239.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::generic_proxy< RTYPE >

Definition at line 30 of file 00_forward_proxy.h.

6.239.2 Member Typedef Documentation

6.239.2.1 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::generic_proxy< RTYPE >::VECTOR

Definition at line 29 of file generic_proxy.h.

6.239.3 Constructor & Destructor Documentation

6.239.3.1 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::generic_proxy ( ) [inline]

Definition at line 31 of file generic_proxy.h.

6.239.3.2 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::generic_proxy (const generic_proxy< RTYPE > & *other*) [inline]

Definition at line 33 of file generic_proxy.h.

6.239.3.3 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::generic_proxy (VECTOR & *v*, R_xlen_t *i*) [inline]

Definition at line 36 of file generic_proxy.h.

6.239.4 Member Function Documentation

6.239.4.1 template<int RTYPE> SEXP Rcpp::internal::generic_proxy< RTYPE >::get ( ) const [inline], [private]

Definition at line 85 of file generic_proxy.h.

6.239.4.2 template<int RTYPE> void Rcpp::internal::generic_proxy< RTYPE >::import (const generic_proxy< RTYPE > & *other*) [inline]

Definition at line 76 of file generic_proxy.h.

6.239.4.3 template<int RTYPE> void Rcpp::internal::generic_proxy< RTYPE >::move (R_xlen_t *n*) [inline]

Definition at line 74 of file generic_proxy.h.

6.239.4.4 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::operator bool ( ) const [inline]

Definition at line 63 of file generic_proxy.h.

6.239.4.5 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::operator int ( ) const [inline]

Definition at line 64 of file generic_proxy.h.

6.239.4.6 template<int RTYPE> Rcpp::internal::generic_proxy< RTYPE >::operator SEXP ( ) const [inline]

Definition at line 54 of file generic_proxy.h.

6.239.4.7 template<int RTYPE> template<typename U > Rcpp::internal::generic_proxy< RTYPE >::operator U ( ) const [inline]

Definition at line 58 of file generic_proxy.h.

6.239.4.8 template<int RTYPE> generic_proxy& Rcpp::internal::generic_proxy< RTYPE >::operator= (SEXP *rhs*) [inline]

Definition at line 38 of file generic_proxy.h.

6.239.4.9 template<int RTYPE> generic_proxy& Rcpp::internal::generic_proxy< RTYPE >::operator= (const generic_proxy< RTYPE > & *rhs*) [inline]

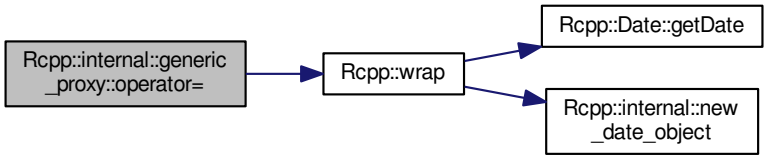
Definition at line 43 of file generic_proxy.h.

6.239.4.10 template<int RTYPE> template<typename T > generic_proxy& Rcpp::internal::generic_proxy< RTYPE >::operator= (const T & *rhs*) [inline]

Definition at line 49 of file generic_proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.239.4.11 template<int RTYPE> void Rcpp::internal::generic_proxy< RTYPE >::set (SEXP *x*) [inline], [private]

Definition at line 82 of file generic_proxy.h.

6.239.4.12 template<int RTYPE> void Rcpp::internal::generic_proxy< RTYPE >::swap (generic_proxy< RTYPE > & *other*) [inline]

Definition at line 66 of file generic_proxy.h.

6.239.5 Member Data Documentation

6.239.5.1 template<int RTYPE> R_xlen_t Rcpp::internal::generic_proxy< RTYPE >::index

Definition at line 73 of file generic_proxy.h.

6.239.5.2 template<int RTYPE> VECTOR* Rcpp::internal::generic_proxy< RTYPE >::parent

Definition at line 72 of file generic_proxy.h.

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

6.240 Rcpp::GenericProxy< Proxy > Struct Template Reference

`#include <GenericProxy.h>`

Public Member Functions

- SEXP get () const

6.240.1 Detailed Description

template<typename Proxy>
struct Rcpp::GenericProxy< Proxy >

Definition at line 24 of file GenericProxy.h.

6.240.2 Member Function Documentation

6.240.2.1 template<typename Proxy> SEXP Rcpp::GenericProxy< Proxy >::get ( ) const [inline]

Definition at line 25 of file GenericProxy.h.

Referenced by Rcpp::Vector< INTSXP >::Vector().

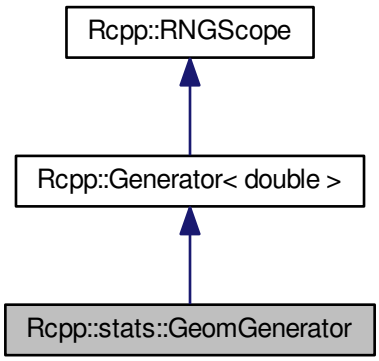
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/proxy/GenericProxy.h

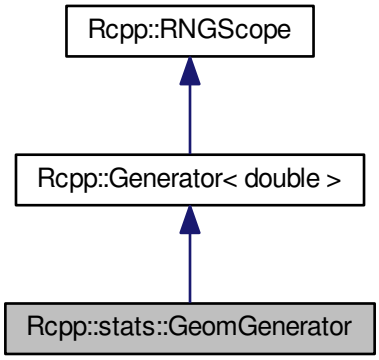
6.241 Rcpp::stats::GeomGenerator Class Reference

`#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

6.241.1 Detailed Description

Definition at line 28 of file rgeom.h.

6.241.2 Constructor & Destructor Documentation

6.241.2.1 Rcpp::stats::GeomGenerator::GeomGenerator (double *p*) [inline]

Definition at line 31 of file rgeom.h.

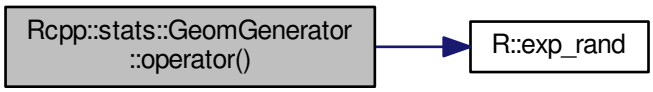
6.241.3 Member Function Documentation

6.241.3.1 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:



6.241.4 Member Data Documentation

6.241.4.1 double Rcpp::stats::GeomGenerator::lambda [private]

Definition at line 38 of file rgeom.h.

Referenced by operator()().

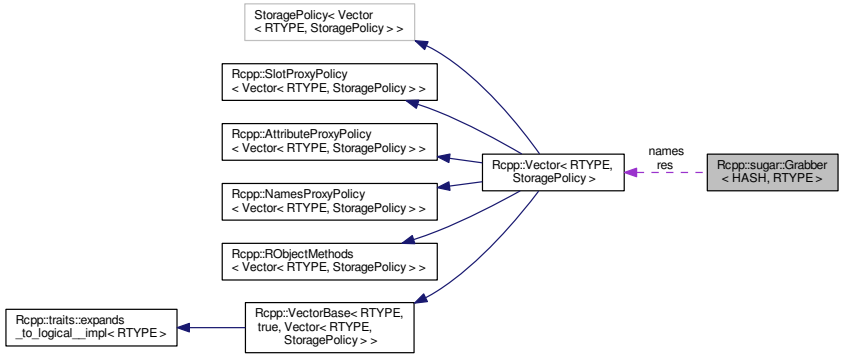
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rgeom.h

6.242 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

6.242.1 Detailed Description

template<typename HASH, int RTYPE>
class Rcpp::sugar::Grabber< HASH, RTYPE >

Definition at line 42 of file table.h.

6.242.2 Constructor & Destructor Documentation

6.242.2.1 template<typename HASH, int RTYPE> Rcpp::sugar::Grabber< HASH, RTYPE >::Grabber (IntegerVector & *res_*, CharacterVector & *names_*) [inline]

Definition at line 44 of file table.h.

6.242.3 Member Function Documentation

6.242.3.1 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.

6.242.4 Member Data Documentation

6.242.4.1 template<typename HASH, int RTYPE> R_xlen_t Rcpp::sugar::Grabber< HASH, RTYPE >::index [private]

Definition at line 55 of file table.h.

6.242.4.2 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::Table< RTYPE, TABLE_T >::operator IntegerVector().

6.242.4.3 template<typename HASH, int RTYPE> IntegerVector& Rcpp::sugar::Grabber< HASH, RTYPE >::res [private]

Definition at line 53 of file table.h.

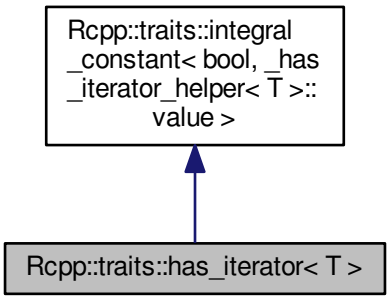
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

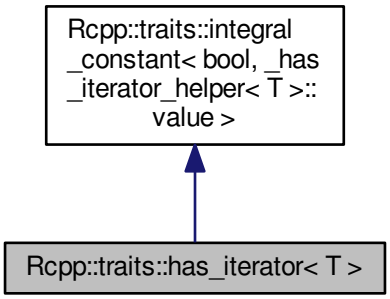
6.243 Rcpp::traits::has_iterator< T > Struct Template Reference

`#include <has_iterator.h>`

Inheritance diagram for Rcpp::traits::has_iterator< T >:



Collaboration diagram for Rcpp::traits::has_iterator< T >:



Additional Inherited Members

6.243.1 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.

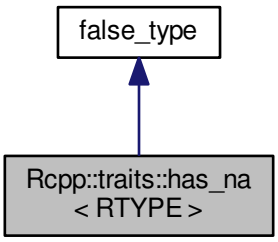
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.244 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

6.244.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::has_na< RTYPE >

Indentifies 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.

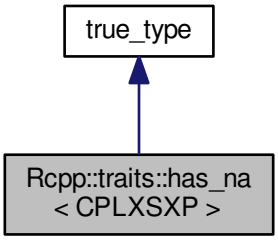
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

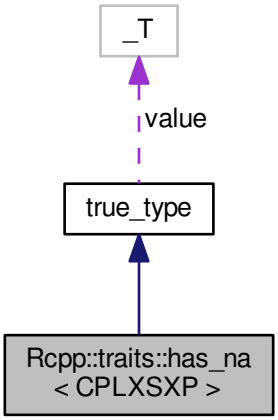
6.245 Rcpp::traits::has_na< CPLXSXP > Struct Template Reference

`#include <has_na.h>`

Inheritance diagram for Rcpp::traits::has_na< CPLXSXP >:



Collaboration diagram for Rcpp::traits::has_na< CPLXSXP >:



Additional Inherited Members

6.245.1 Detailed Description

template<>
struct Rcpp::traits::has_na< CPLXSXP >

complex vectors support missing values

Definition at line 50 of file has_na.h.

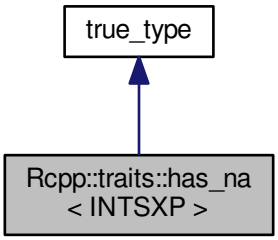
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

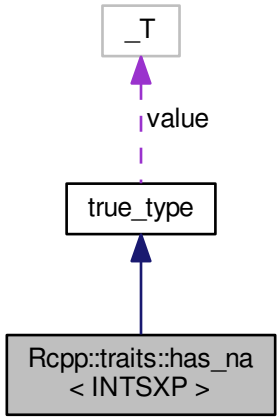
6.246 Rcpp::traits::has_na< INTSXP > Struct Template Reference

`#include <has_na.h>`

Inheritance diagram for Rcpp::traits::has_na< INTSXP >:



Collaboration diagram for Rcpp::traits::has_na< INTSXP >:



Additional Inherited Members

6.246.1 Detailed Description

template<>
struct Rcpp::traits::has_na< INTSXP >

integer vectors support missing values

Definition at line 40 of file has_na.h.

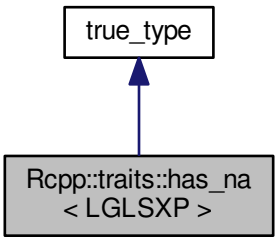
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

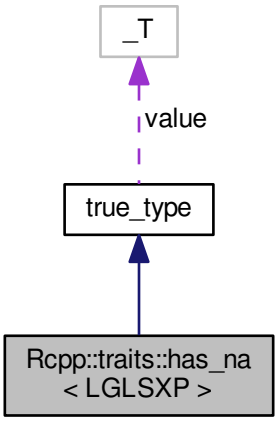
6.247 Rcpp::traits::has_na< LGLSXP > Struct Template Reference

`#include <has_na.h>`

Inheritance diagram for Rcpp::traits::has_na< LGLSXP >:



Collaboration diagram for Rcpp::traits::has_na< LGLSXP >:



Additional Inherited Members

6.247.1 Detailed Description

template<>
struct Rcpp::traits::has_na< LGLSXP >

logical vectors support missing values

Definition at line 60 of file has_na.h.

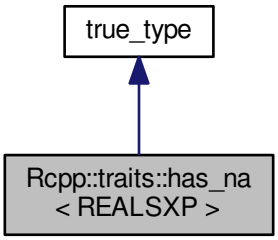
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

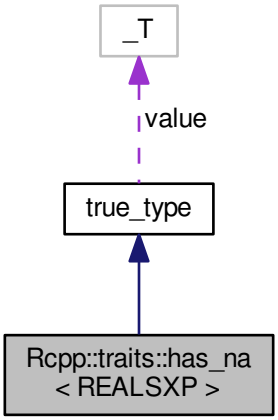
6.248 Rcpp::traits::has_na< REALSXP > Struct Template Reference

`#include <has_na.h>`

Inheritance diagram for Rcpp::traits::has_na< REALSXP >:



Collaboration diagram for Rcpp::traits::has_na< REALSXP >:



Additional Inherited Members

6.248.1 Detailed Description

template<>
struct Rcpp::traits::has_na< REALSXP >

numeric vectors support missing values

Definition at line 45 of file has_na.h.

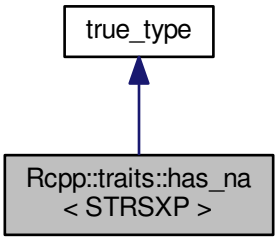
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

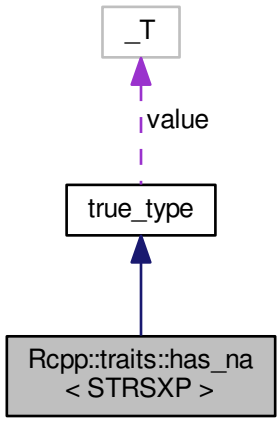
6.249 Rcpp::traits::has_na< STRSXP > Struct Template Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< STRSXP >:



Collaboration diagram for Rcpp::traits::has_na< STRSXP >:



Additional Inherited Members

6.249.1 Detailed Description

template<>
struct Rcpp::traits::has_na< STRSXP >

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

6.250 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 (...)

6.250.1 Detailed Description

template<typename T>
class Rcpp::sugar::cbind_impl::detail::has_stored_type< T >

Definition at line 373 of file cbind.h.

6.250.2 Member Typedef Documentation

6.250.2.1 template<typename T > typedef char Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::yes [private]

Definition at line 375 of file cbind.h.

6.250.3 Member Function Documentation

6.250.3.1 template<typename T > template<typename C > static yes Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (typename C::stored_type *) [static], [private]

6.250.3.2 template<typename T > template<typename C > static no Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (...) [static], [private]

6.250.4 Member Data Documentation

6.250.4.1 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.

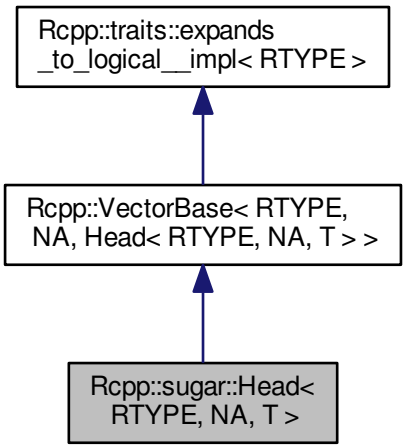
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

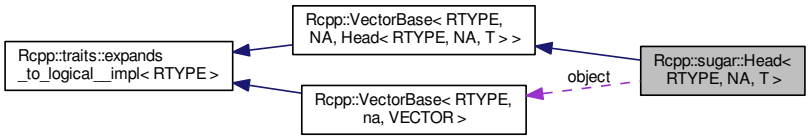
6.251 Rcpp::sugar::Head< RTYPE, NA, T > Class Template Reference

`#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

6.251.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Head< RTYPE, NA, T >

Definition at line 29 of file head.h.

6.251.2 Member Typedef Documentation

6.251.2.1 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.

6.251.2.2 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.

6.251.3 Constructor & Destructor Documentation

6.251.3.1 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.

6.251.4 Member Function Documentation

6.251.4.1 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.

6.251.4.2 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.

6.251.5 Member Data Documentation

6.251.5.1 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(), Rcpp::head(), and Rcpp::sugar::Head< RTYPE, NA, T >::size().

6.251.5.2 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.

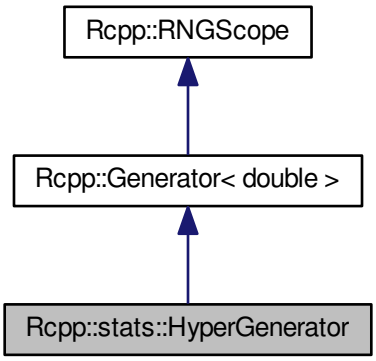
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/head.h

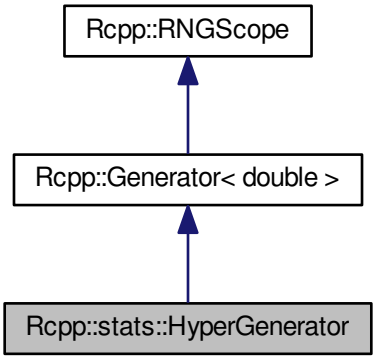
6.252 Rcpp::stats::HyperGenerator Class Reference

`#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

6.252.1 Detailed Description

Definition at line 28 of file rhyper.h.

6.252.2 Constructor & Destructor Documentation

6.252.2.1 Rcpp::stats::HyperGenerator::HyperGenerator (double *nn1_*, double *nn2_*, double *kk_*) [inline]

Definition at line 30 of file rhyper.h.

6.252.3 Member Function Documentation

6.252.3.1 double Rcpp::stats::HyperGenerator::operator() ( ) const [inline]

Definition at line 32 of file rhyper.h.

References kk, nn1, and nn2.

6.252.4 Member Data Documentation

6.252.4.1 double Rcpp::stats::HyperGenerator::kk [private]

Definition at line 34 of file rhyper.h.

Referenced by operator()().

6.252.4.2 double Rcpp::stats::HyperGenerator::nn1 [private]

Definition at line 34 of file rhyper.h.

Referenced by operator()().

6.252.4.3 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

6.253 Rcpp::traits::identity< T > Struct Template Reference

`#include <traits.h>`

Public Types

- typedef T type

6.253.1 Detailed Description

template<typename T>
struct Rcpp::traits::identity< T >

Definition at line 30 of file traits.h.

6.253.2 Member Typedef Documentation

6.253.2.1 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

6.254 Rcpp::traits::if_< COND, LHS, RHS > Struct Template Reference

`#include <if_.h>`

Public Types

- typedef LHS type

6.254.1 Detailed Description

template<bool COND, typename LHS, typename RHS>
struct Rcpp::traits::if_< COND, LHS, RHS >

Definition at line 29 of file if_.h.

6.254.2 Member Typedef Documentation

6.254.2.1 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

6.255 Rcpp::traits::if_< false, LHS, RHS > Struct Template Reference

`#include <if_.h>`

Public Types

- typedef RHS type

6.255.1 Detailed Description

template<typename LHS, typename RHS>
struct Rcpp::traits::if_< false, LHS, RHS >

Definition at line 34 of file if_.h.

6.255.2 Member Typedef Documentation

6.255.2.1 template<typename LHS , typename RHS > typedef RHS Rcpp::traits::if_< false, LHS, RHS >::type

Definition at line 35 of file if_.h.

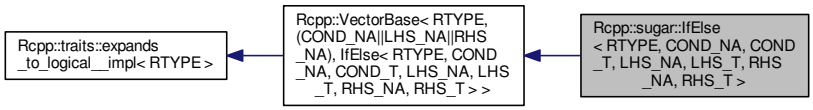
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/if_.h

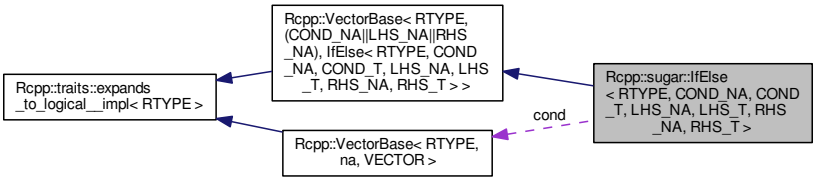
6.256 Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.256.1 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.

6.256.2 Member Typedef Documentation

6.256.2.1 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.

6.256.2.2 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.

6.256.2.3 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.

6.256.2.4 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.

6.256.3 Constructor & Destructor Documentation

6.256.3.1 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.

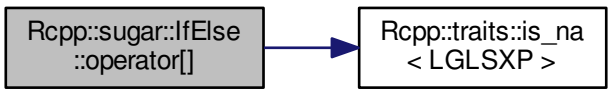
6.256.4 Member Function Documentation

6.256.4.1 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:

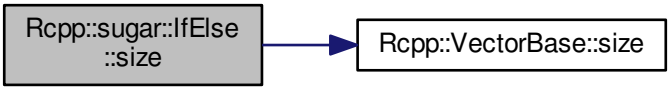


6.256.4.2 template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T> R_xlen_t Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const [inline]

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:



6.256.5 Member Data Documentation

6.256.5.1 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::ifelse(), 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_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_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_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[](), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[](), 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(), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::size().

6.256.5.2 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::ifelse(), 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_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_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_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[](), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[]().

6.256.5.3 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::ifelse(), 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_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_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_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[](), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[]().

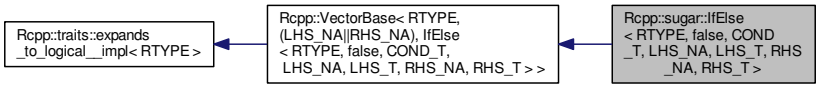
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

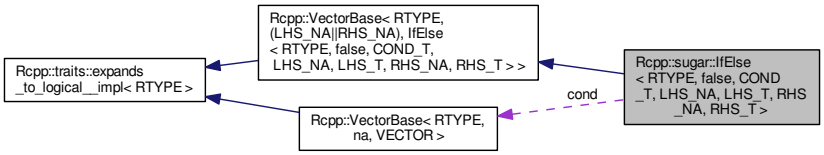
6.257 Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.257.1 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_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 77 of file ifelse.h.

6.257.2 Member Typedef Documentation

6.257.2.1 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.

6.257.2.2 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.

6.257.2.3 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.

6.257.2.4 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.

6.257.2.5 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.

6.257.2.6 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.

6.257.3 Constructor & Destructor Documentation

6.257.3.1 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.

6.257.4 Member Function Documentation

6.257.4.1 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.

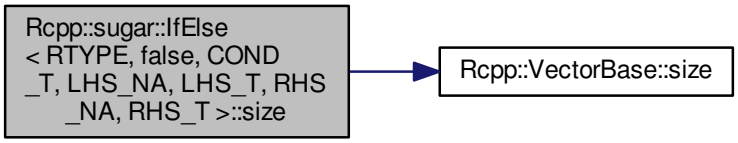
References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.257.4.2 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:



6.257.5 Member Data Documentation

6.257.5.1 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.

6.257.5.2 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.

6.257.5.3 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.

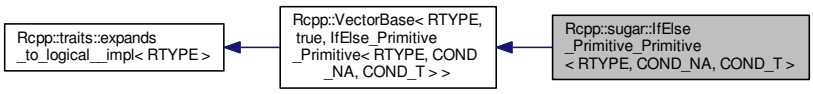
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

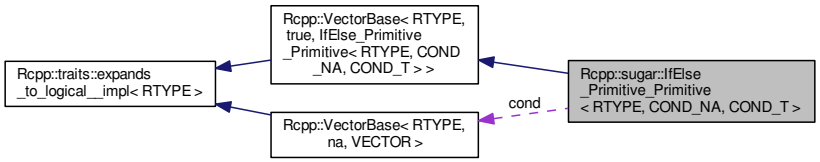
6.258 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > Class Template Reference

`#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

6.258.1 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.

6.258.2 Member Typedef Documentation

6.258.2.1 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.

6.258.2.2 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.

6.258.3 Constructor & Destructor Documentation

6.258.3.1 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.

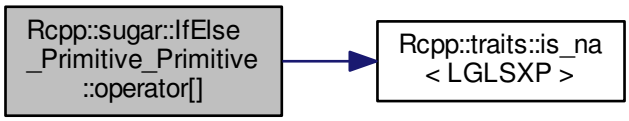
6.258.4 Member Function Documentation

6.258.4.1 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< 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:

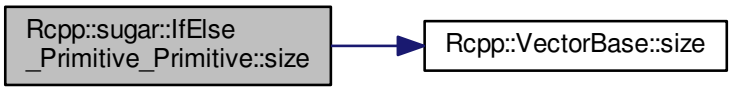


6.258.4.2 template<int RTYPE, bool COND_NA, typename COND_T> R_xlen_t Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::size ( ) const [inline]

Definition at line 293 of file ifelse.h.

References Rcpp::sugar::IfElse< 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 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.

6.258.5.2 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.

6.258.5.3 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.

6.258.5.4 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.

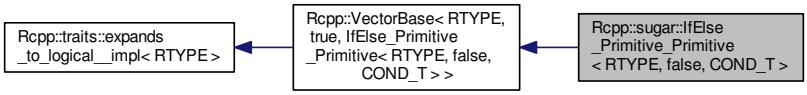
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

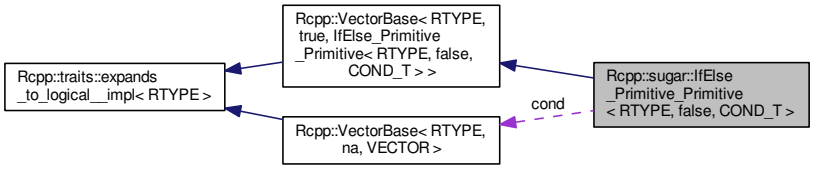
6.259 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T > Class Template Reference

`#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

6.259.1 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.

6.259.2 Member Typedef Documentation

6.259.2.1 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.

6.259.2.2 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.

6.259.3 Constructor & Destructor Documentation

6.259.3.1 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.

6.259.4 Member Function Documentation

6.259.4.1 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.

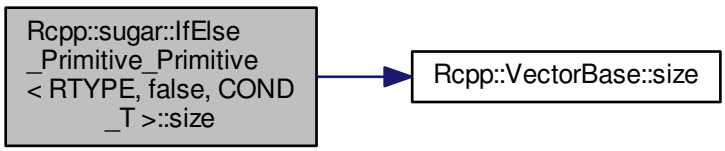
References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.259.4.2 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< 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 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.

6.259.5.2 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.

6.259.5.3 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.

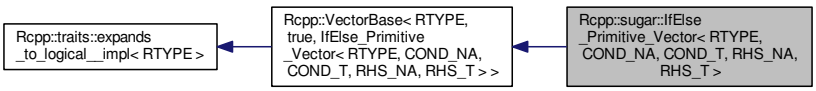
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

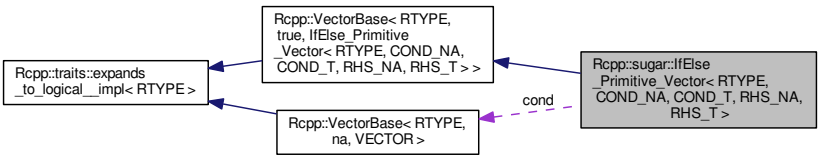
6.260 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.260.1 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.

6.260.2 Member Typedef Documentation

6.260.2.1 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.

6.260.2.2 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.

6.260.2.3 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.

6.260.2.4 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.

6.260.3 Constructor & Destructor Documentation

6.260.3.1 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.

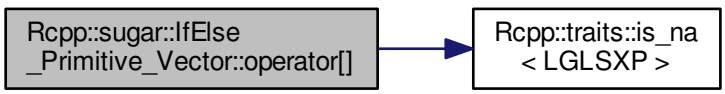
6.260.4 Member Function Documentation

6.260.4.1 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< 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:

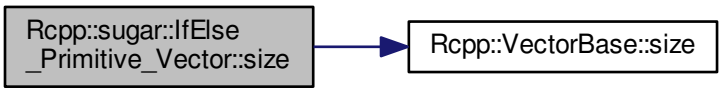


6.260.4.2 template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T> R_xlen_t Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size ( ) const [inline]

Definition at line 143 of file ifelse.h.

References Rcpp::sugar::IfElse< 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:



6.260.5 Member Data Documentation

6.260.5.1 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.

6.260.5.2 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.

6.260.5.3 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.

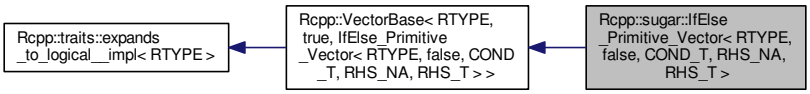
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

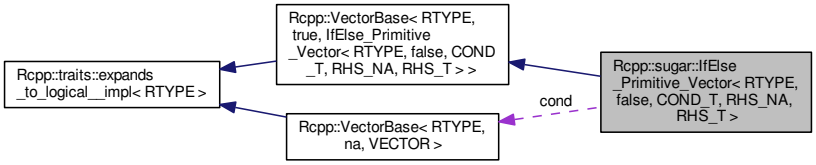
6.261 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.261.1 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.

6.261.2 Member Typedef Documentation

6.261.2.1 template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<L↩ GLSXP,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.

6.261.2.2 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.

6.261.2.3 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.

6.261.2.4 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.

6.261.3 Constructor & Destructor Documentation

6.261.3.1 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.

6.261.4 Member Function Documentation

6.261.4.1 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.

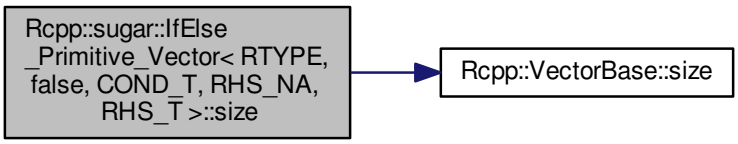
References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.261.4.2 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< 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:



6.261.5 Member Data Documentation

6.261.5.1 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.

6.261.5.2 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.

6.261.5.3 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.

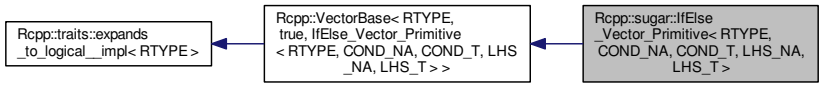
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

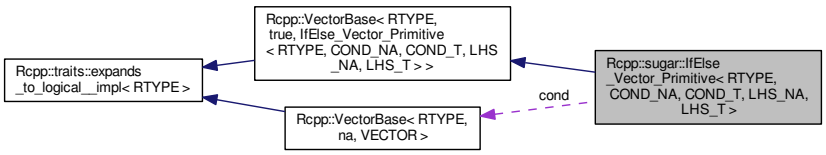
6.262 Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > Class Template Reference

`#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

6.262.1 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.

6.262.2 Member Typedef Documentation

6.262.2.1 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.

6.262.2.2 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.

6.262.2.3 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.

6.262.2.4 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.

6.262.3 Constructor & Destructor Documentation

6.262.3.1 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.

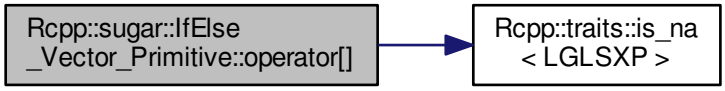
6.262.4 Member Function Documentation

6.262.4.1 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< 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:



6.262.4.2 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< 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:

6.262.5 Member Data Documentation

6.262.5.1 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.

6.262.5.2 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.

6.262.5.3 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.

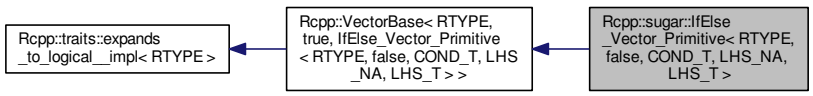
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

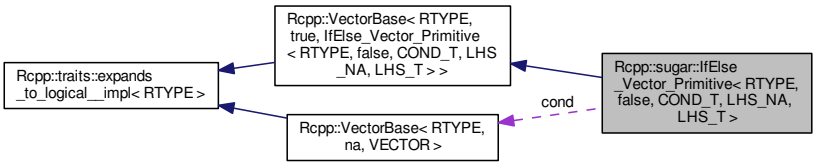
6.263 Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > Class Template Reference

`#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

6.263.1 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.

6.263.2 Member Typedef Documentation

6.263.2.1 template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T > typedef Rcpp::VectorBase<LG↩ LSXP,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.

6.263.2.2 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.

6.263.2.3 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.

6.263.2.4 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.

6.263.3 Constructor & Destructor Documentation

6.263.3.1 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.

6.263.4 Member Function Documentation

6.263.4.1 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.

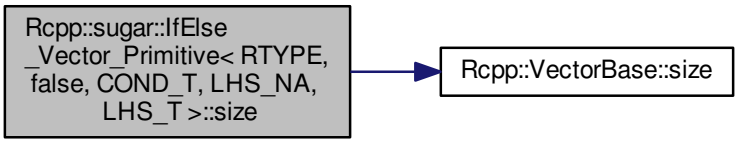
References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.263.4.2 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< 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:



6.263.5 Member Data Documentation

6.263.5.1 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.

6.263.5.2 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.

6.263.5.3 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.

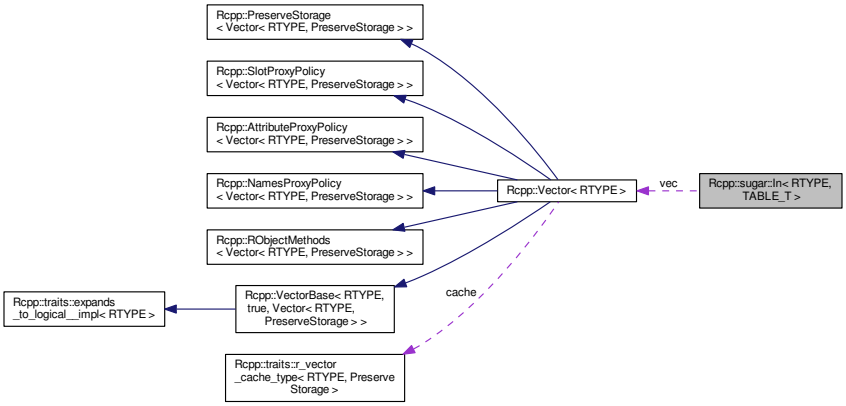
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

6.264 Rcpp::sugar::In< RTYPE, TABLE_T > Class Template Reference

```
#include <unique.h>
```

Collaboration diagram for Rcpp::sugar::In< RTYPE, TABLE_T >:



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

6.264.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::In< RTYPE, TABLE_T >

Definition at line 44 of file unique.h.

6.264.2 Member Typedef Documentation

6.264.2.1 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.

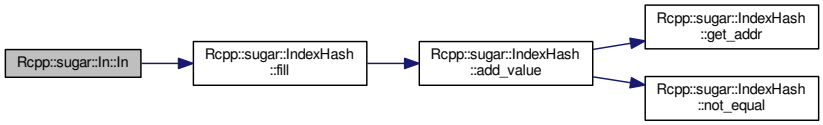
6.264.3 Constructor & Destructor Documentation

6.264.3.1 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().

Here is the call graph for this function:



6.264.4 Member Function Documentation

6.264.4.1 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::InSet< HASH >::hash.

6.264.5 Member Data Documentation

6.264.5.1 template<int RTYPE, typename TABLE_T > HASH Rcpp::sugar::In< RTYPE, TABLE_T >::hash [private]

Definition at line 47 of file unique.h.

6.264.5.2 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

6.265 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)
- int get_index (STORAGE value) const
- int get_addr (STORAGE value) const
- template<>
  int get_addr (int value) const
- template<>
  int get_addr (double val) const
- template<>
  int get_addr (SEXP value) const

Public Attributes

- int n
- int m
- int k
- STORAGE * src
- int size_
- int * data

6.265.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::IndexHash< RTYPE >

Definition at line 50 of file IndexHash.h.

6.265.2 Member Typedef Documentation

6.265.2.1 template<int RTYPE> typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IndexHash< RTYPE >::STORAGE

Definition at line 52 of file IndexHash.h.

6.265.2.2 template<int RTYPE> typedef Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::VECTOR

Definition at line 53 of file IndexHash.h.

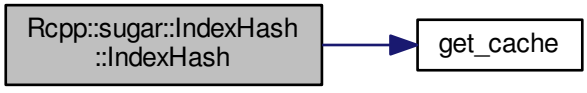
6.265.3 Constructor & Destructor Documentation

6.265.3.1 template<int RTYPE> Rcpp::sugar::IndexHash< RTYPE >::IndexHash (SEXP *table*) [inline]

Definition at line 55 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:



6.265.4 Member Function Documentation

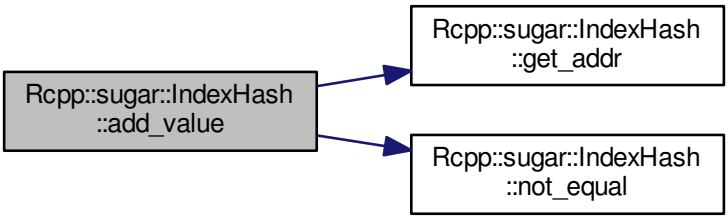
6.265.4.1 template<int RTYPE> bool Rcpp::sugar::IndexHash< RTYPE >::add_value (int *i*) [inline]

Definition at line 167 of file IndexHash.h.

References DEMANGLE, Rcpp::sugar::IndexHash< RTYPE >::get_addr(), Rcpp::sugar::IndexHash< RTYPE >::not_equal(), and RCPP_DEBUG_2.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), and Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated().

Here is the call graph for this function:

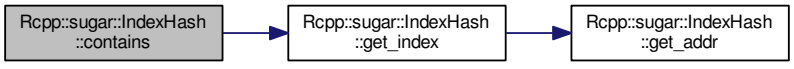


6.265.4.2 template<int RTYPE> bool Rcpp::sugar::IndexHash< RTYPE >::contains (STORAGE *val*) const [inline]

Definition at line 102 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get_index().

Here is the call graph for this function:



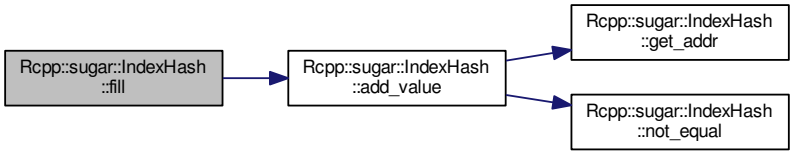
6.265.4.3 template<int RTYPE> IndexHash& Rcpp::sugar::IndexHash< RTYPE >::fill ( ) [inline]

Definition at line 74 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::n, RCPP_PRO↩ FILE_RECORD, RCPP_PROFILE_TIC, and RCPP_PROFILE_TOC.

Referenced by Rcpp::sugar::In< RTYPE, TABLE_T >::In(), and Rcpp::unique().

Here is the call graph for this function:



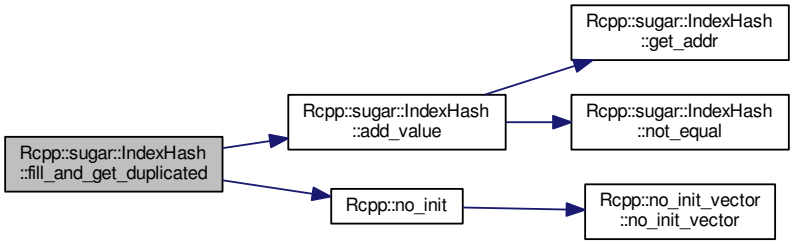
6.265.4.4 template<int RTYPE> LogicalVector Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated ( ) [inline]

Definition at line 85 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:



6.265.4.5 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::get_addr (STORAGE *value*) const

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), and Rcpp::sugar::IndexHash< RTYPE >::get_↩ index().

6.265.4.6 template<> int Rcpp::sugar::IndexHash< INTSXP >::get_addr (int *value*) const [inline]

Definition at line 204 of file IndexHash.h.

References RCPP_HASH.

6.265.4.7 template<> int Rcpp::sugar::IndexHash< REALSXP >::get_addr (double *val*) const [inline]

Definition at line 208 of file IndexHash.h.

References RCPP_HASH.

6.265.4.8 template<> int Rcpp::sugar::IndexHash< STRSXP >::get_addr (SEXP *value*) const [inline]

Definition at line 225 of file IndexHash.h.

References RCPP_HASH.

6.265.4.9 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::get_index (STORAGE *value*) const [inline]

Definition at line 188 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get_addr().

Referenced by Rcpp::sugar::IndexHash< RTYPE >::contains(), and Rcpp::sugar::IndexHash< RTYPE >::lookup__↩ impl().

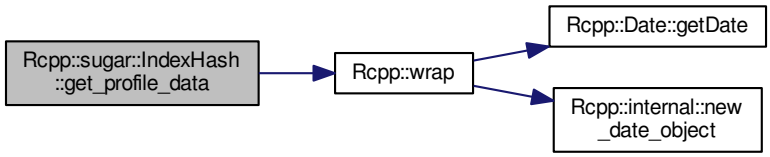
Here is the call graph for this function:

6.265.4.10 template<int RTYPE> SEXP Rcpp::sugar::IndexHash< RTYPE >::get_profile_data ( ) [inline]

Definition at line 155 of file IndexHash.h.

References Rcpp::wrap().

Here is the call graph for this function:



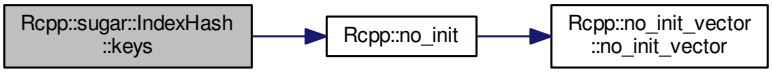
6.265.4.11 template<int RTYPE> Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::keys ( ) const [inline]

Definition at line 111 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:



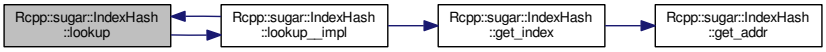
6.265.4.12 template<int RTYPE> template<typename T > SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (const T & *vec*) const [inline]

Definition at line 93 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:

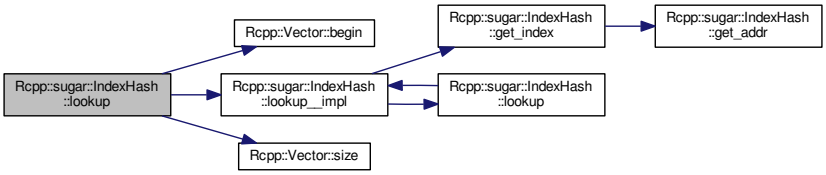


6.265.4.13 template<int RTYPE> SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (const VECTOR & *vec*) const [inline]

Definition at line 98 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:



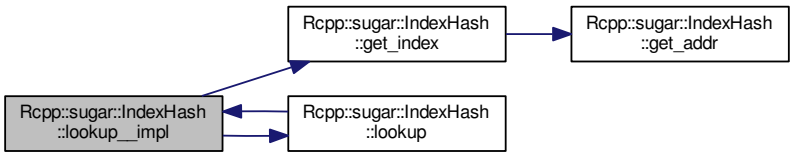
6.265.4.14 template<int RTYPE> template<typename T > SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup__impl (const T & *vec*, int *n_*) const [inline]

Definition at line 135 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:



6.265.4.15 template<int RTYPE> bool Rcpp::sugar::IndexHash< RTYPE >::not_equal (const STORAGE & *lhs,* const STORAGE & *rhs*) [inline]

Definition at line 163 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value().

6.265.4.16 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::size ( ) const [inline]

Definition at line 106 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::size_.

6.265.5 Member Data Documentation

6.265.5.1 template<int RTYPE> int* Rcpp::sugar::IndexHash< RTYPE >::data

Definition at line 123 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

6.265.5.2 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::k

Definition at line 119 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

6.265.5.3 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::m

Definition at line 119 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

6.265.5.4 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::n

Definition at line 119 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().

6.265.5.5 template<int RTYPE> int Rcpp::sugar::IndexHash< RTYPE >::size_

Definition at line 121 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::keys(), and Rcpp::sugar::IndexHash< RTYPE >::size().

6.265.5.6 template<int RTYPE> STORAGE* Rcpp::sugar::IndexHash< RTYPE >::src

Definition at line 120 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::keys().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/hash/IndexHash.h

6.266 Rcpp::traits::init_type< RTYPE > Struct Template Reference

`#include <init_type.h>`

Public Types

- typedef storage_type< RTYPE >::type type

6.266.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::init_type< RTYPE >

Definition at line 28 of file init_type.h.

6.266.2 Member Typedef Documentation

6.266.2.1 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

6.267 Rcpp::traits::init_type< LGLSXP > Struct Template Reference

`#include <init_type.h>`

Public Types

- typedef bool type

6.267.1 Detailed Description

template<>
struct Rcpp::traits::init_type< LGLSXP >

Definition at line 34 of file init_type.h.

6.267.2 Member Typedef Documentation

6.267.2.1 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

6.268 Rcpp::traits::init_type< STRSXP > Struct Template Reference

`#include <init_type.h>`

Public Types

- typedef const char * type

6.268.1 Detailed Description

template<>
struct Rcpp::traits::init_type< STRSXP >

Definition at line 31 of file init_type.h.

6.268.2 Member Typedef Documentation

6.268.2.1 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

6.269 Rcpp::traits::input_parameter< T > Struct Template Reference

`#include <InputParameter.h>`

Public Types

- typedef Rcpp::InputParameter< T > type

6.269.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< T >

Definition at line 82 of file InputParameter.h.

6.269.2 Member Typedef Documentation

6.269.2.1 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

6.270 Rcpp::traits::input_parameter< const T & > Struct Template Reference

`#include <InputParameter.h>`

Public Types

- typedef Rcpp::ConstReferenceInputParameter< T > type

6.270.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< const T & >

Definition at line 94 of file InputParameter.h.

6.270.2 Member Typedef Documentation

6.270.2.1 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

6.271 Rcpp::traits::input_parameter< const T > Struct Template Reference

`#include <InputParameter.h>`

Public Types

- typedef Rcpp::ConstInputParameter< T > type

6.271.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< const T >

Definition at line 90 of file InputParameter.h.

6.271.2 Member Typedef Documentation

6.271.2.1 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

6.272 Rcpp::traits::input_parameter< T & > Struct Template Reference

`#include <InputParameter.h>`

Public Types

- typedef Rcpp::ReferenceInputParameter< T > type

6.272.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< T & >

Definition at line 86 of file InputParameter.h.

6.272.2 Member Typedef Documentation

6.272.2.1 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

6.273 Rcpp::InputParameter< T > Class Template Reference

`#include <InputParameter.h>`

Public Member Functions

- InputParameter (SEXP x_)
- operator T ()

Private Attributes

- SEXP x

6.273.1 Detailed Description

template<typename T>
class Rcpp::InputParameter< T >

Definition at line 30 of file InputParameter.h.

6.273.2 Constructor & Destructor Documentation

6.273.2.1 template<typename T > Rcpp::InputParameter< T >::InputParameter (SEXP x_) [inline]

Definition at line 32 of file InputParameter.h.

6.273.3 Member Function Documentation

6.273.3.1 template<typename T > Rcpp::InputParameter< T >::operator T ( ) [inline]

Definition at line 34 of file InputParameter.h.

References Rcpp::InputParameter< T >::x.

6.273.4 Member Data Documentation

6.273.4.1 template<typename T > SEXP 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

6.274 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

6.274.1 Detailed Description

template<typename HASH>
class Rcpp::sugar::InSet< HASH >

Definition at line 29 of file unique.h.

6.274.2 Member Typedef Documentation

6.274.2.1 template<typename HASH > typedef HASH::STORAGE Rcpp::sugar::InSet< HASH >::STORAGE [private]

Definition at line 30 of file unique.h.

6.274.3 Constructor & Destructor Documentation

6.274.3.1 template<typename HASH > Rcpp::sugar::InSet< HASH >::InSet (const HASH & *hash_*) [inline]

Definition at line 33 of file unique.h.

6.274.4 Member Function Documentation

6.274.4.1 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.

6.274.5 Member Data Documentation

6.274.5.1 template<typename HASH > const HASH& Rcpp::sugar::InSet< HASH >::hash [private]

Definition at line 40 of file unique.h.

Referenced by Rcpp::sugar::In< RTYPE, TABLE_T >::get(), Rcpp::sugar::InSet< HASH >::operator()(), and Rcpp↩
::unique().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/unique.h

6.275 Rcpp::traits::int2type< I > Struct Template Reference

`#include <traits.h>`

Public Types

- enum { value = I }

6.275.1 Detailed Description

template<int I>
struct Rcpp::traits::int2type< I >

Definition at line 33 of file traits.h.

6.275.2 Member Enumeration Documentation

6.275.2.1 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:

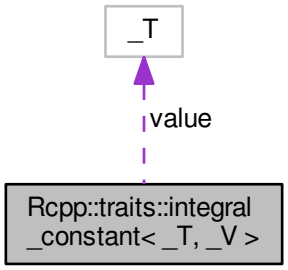
- inst/include/Rcpp/traits/traits.h

6.276 Rcpp::traits::integral_constant< _T, _V > Struct Template Reference

`#include <integral_constant.h>`

Inherited by 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< COMPLEXSXP >, 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 short >, 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< named_object< T > >, Rcpp::traits::is_named< Rcpp::Argument >, 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_needscast< T >, Rcpp::traits::r_sexptype_needscast< double >, Rcpp::traits::r_sexptype_needscast< int >, Rcpp::traits::r_sexptype_needscast< Rbyte >, Rcpp::traits::r_sexptype_needscast< Rcomplex >, Rcpp::traits::same_type< T, U >, Rcpp::traits::same_type< T, T >, Rcpp::traits::same_type< r_type_traits< T >::r_category, r_type_primitive_tag >, and Rcpp::traits::same_type< T::value_type, wchar_t >.

Collaboration diagram for Rcpp::traits::integral_constant< _T, _V >:



Public Types

- typedef _T value_type
- typedef integral_constant< _T, _V > type

Static Public Attributes

- static const _T value = _V

6.276.1 Detailed Description

template<typename _T, _T _V>
struct Rcpp::traits::integral_constant< _T, _V >

Definition at line 29 of file integral_constant.h.

6.276.2 Member Typedef Documentation

6.276.2.1 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.

6.276.2.2 template<typename _T, _T _V> typedef _T Rcpp::traits::integral_constant< _T, _V >::value_type

Definition at line 31 of file integral_constant.h.

6.276.3 Member Data Documentation

6.276.3.1 template<typename _T, _T _V> const _T Rcpp::traits::integral_constant< _T, _V >::value = _V [static]

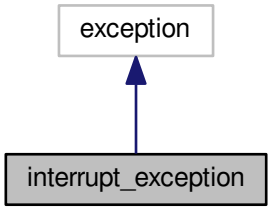
Definition at line 30 of file integral_constant.h.

The documentation for this struct was generated from the following file:

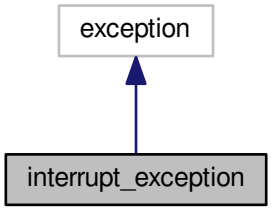
- inst/include/Rcpp/traits/integral_constant.h

6.277 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

6.277.1 Detailed Description

Base class for interrupt exceptions thrown when user interrupts are detected.

Definition at line 15 of file piWithInterrupts.cpp.

6.277.2 Constructor & Destructor Documentation

6.277.2.1 interrupt_exception::interrupt_exception (std::string *message*) [inline]

Constructor.

Parameters

in	*message*	A description of event that caused this exception.

Definition at line 22 of file piWithInterrupts.cpp.

Referenced by PiLeibniz().

6.277.2.2 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.

6.277.3 Member Function Documentation

6.277.3.1 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.

6.277.4 Member Data Documentation

6.277.4.1 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

6.278 Rcpp::internal::InterruptedException Class Reference

`#include <Interrupt.h>`

6.278.1 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

6.279 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

6.279.1 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.

6.279.2 Member Typedef Documentation

6.279.2.1 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.

6.279.2.2 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.

6.279.2.3 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.

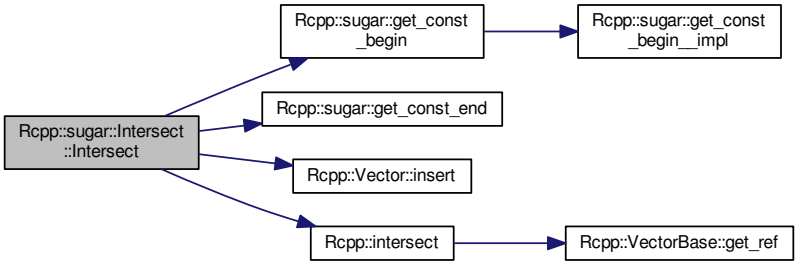
6.279.3 Constructor & Destructor Documentation

6.279.3.1 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(), Rcpp::Vector< RTYPE, StoragePolicy >::insert(), and Rcpp::intersect().

Here is the call graph for this function:



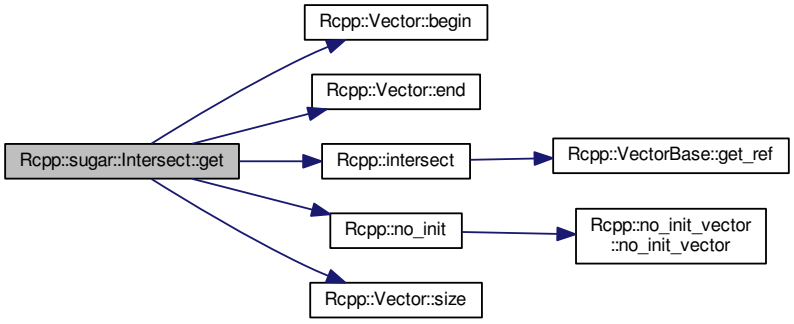
6.279.4 Member Function Documentation

6.279.4.1 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::Vector< RTYPE, StoragePolicy >::end(), Rcpp::intersect(), Rcpp::no_init(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.279.5 Member Data Documentation

6.279.5.1 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.

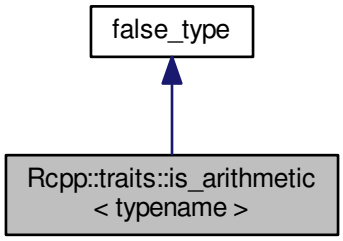
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

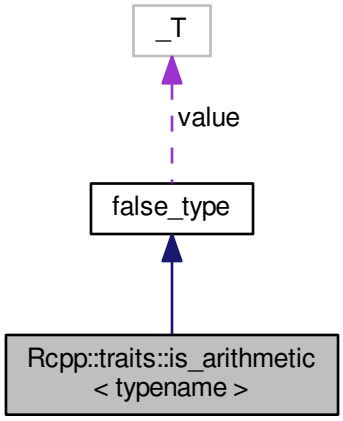
6.280 Rcpp::traits::is_arithmetic< typename > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< typename >:



Collaboration diagram for Rcpp::traits::is_arithmetic< typename >:



Additional Inherited Members

6.280.1 Detailed Description

template<typename>
struct Rcpp::traits::is_arithmetic< typename >

Definition at line 30 of file is_arithmetic.h.

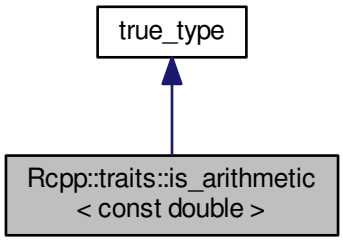
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

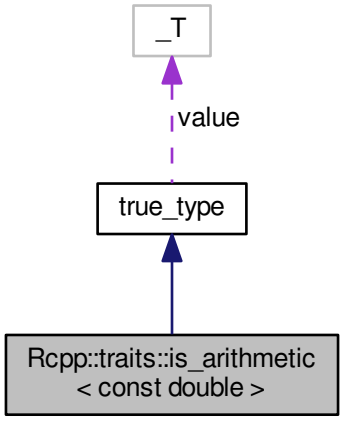
6.281 Rcpp::traits::is_arithmetic< const double > Struct Template Reference

`#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

6.281.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const double >

Definition at line 94 of file is_arithmetic.h.

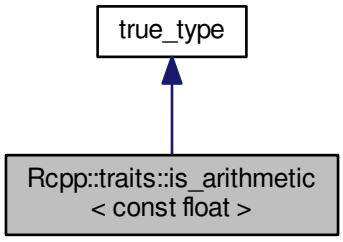
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

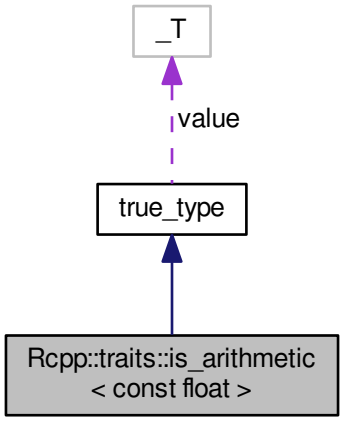
6.282 Rcpp::traits::is_arithmetic< const float > Struct Template Reference

`#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

6.282.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const float >

Definition at line 88 of file is_arithmetic.h.

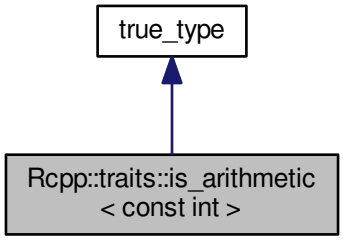
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

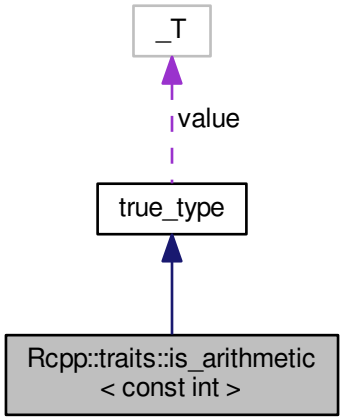
6.283 Rcpp::traits::is_arithmetic< const int > Struct Template Reference

`#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

6.283.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const int >

Definition at line 48 of file is_arithmetic.h.

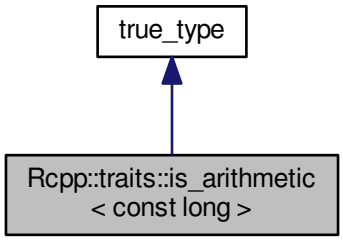
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

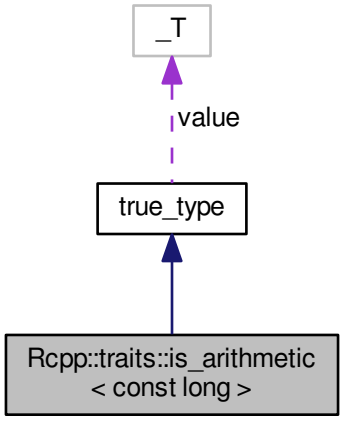
6.284 Rcpp::traits::is_arithmetic< const long > Struct Template Reference

`#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

6.284.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const long >

Definition at line 60 of file is_arithmetic.h.

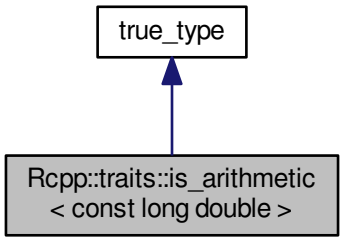
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

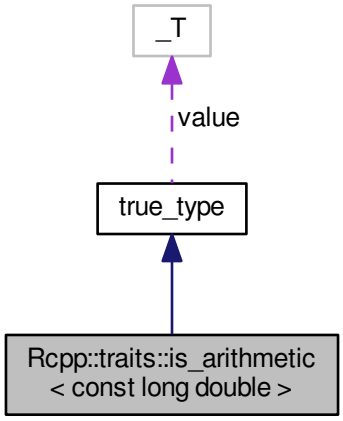
6.285 Rcpp::traits::is_arithmetic< const long double > Struct Template Reference

`#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

6.285.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const long double >

Definition at line 100 of file is_arithmetic.h.

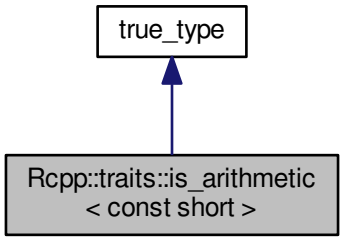
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

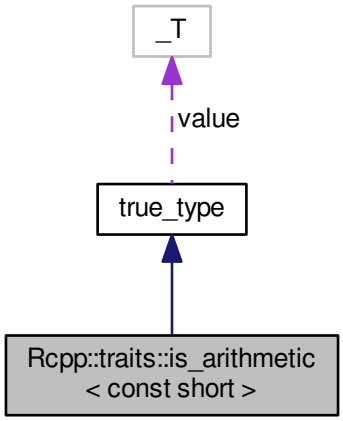
6.286 Rcpp::traits::is_arithmetic< const short > Struct Template Reference

`#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

6.286.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const short >

Definition at line 36 of file is_arithmetic.h.

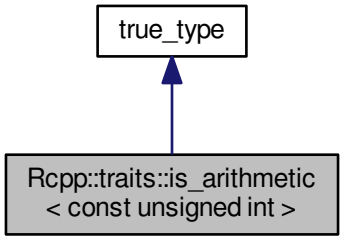
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

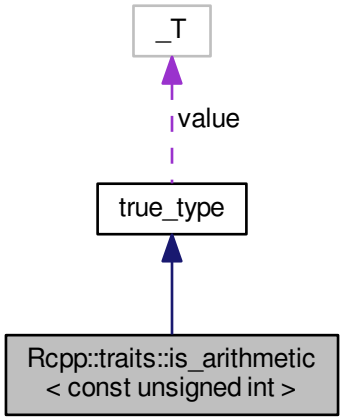
6.287 Rcpp::traits::is_arithmetic< const unsigned int > Struct Template Reference

`#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

6.287.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const unsigned int >

Definition at line 54 of file is_arithmetic.h.

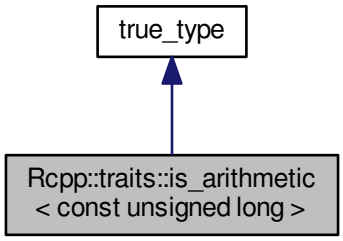
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

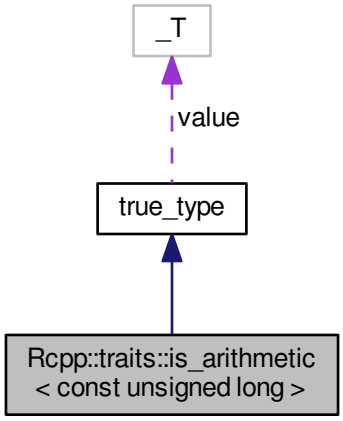
6.288 Rcpp::traits::is_arithmetic< const unsigned long > Struct Template Reference

`#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

6.288.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const unsigned long >

Definition at line 66 of file is_arithmetic.h.

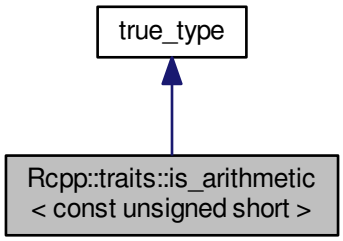
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

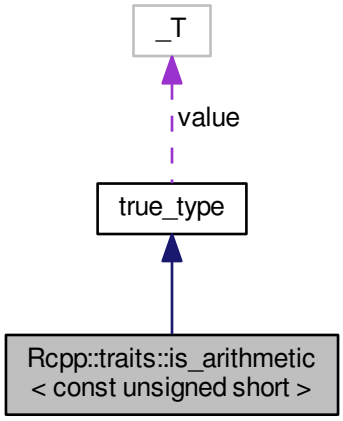
6.289 Rcpp::traits::is_arithmetic< const unsigned short > Struct Template Reference

`#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

6.289.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< const unsigned short >

Definition at line 42 of file is_arithmetic.h.

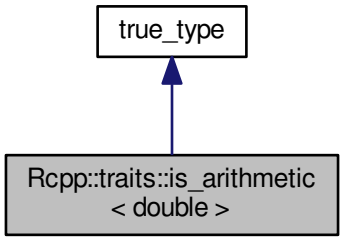
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

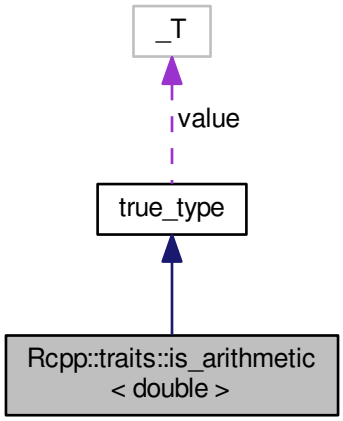
6.290 Rcpp::traits::is_arithmetic< double > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< double >:



Collaboration diagram for Rcpp::traits::is_arithmetic< double >:



Additional Inherited Members

6.290.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< double >

Definition at line 91 of file is_arithmetic.h.

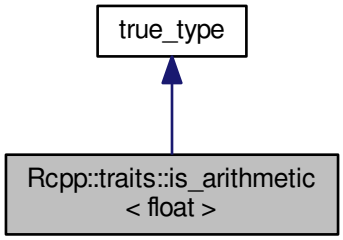
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

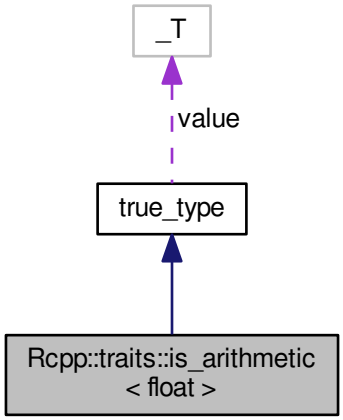
6.291 Rcpp::traits::is_arithmetic< float > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< float >:



Collaboration diagram for Rcpp::traits::is_arithmetic< float >:



Additional Inherited Members

6.291.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< float >

Definition at line 85 of file is_arithmetic.h.

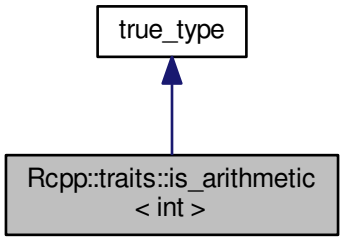
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

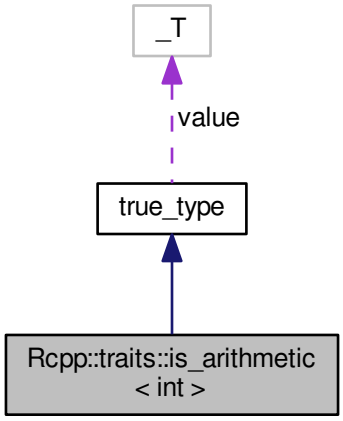
6.292 Rcpp::traits::is_arithmetic< int > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< int >:



Collaboration diagram for Rcpp::traits::is_arithmetic< int >:



Additional Inherited Members

6.292.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< int >

Definition at line 45 of file is_arithmetic.h.

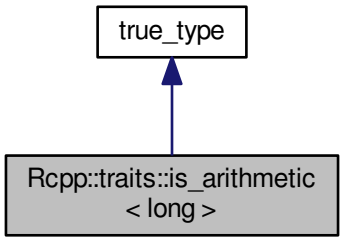
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

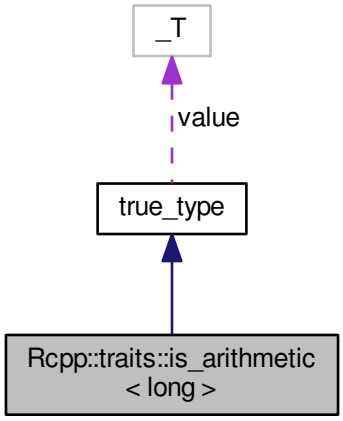
6.293 Rcpp::traits::is_arithmetic< long > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< long >:



Collaboration diagram for Rcpp::traits::is_arithmetic< long >:



Additional Inherited Members

6.293.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< long >

Definition at line 57 of file is_arithmetic.h.

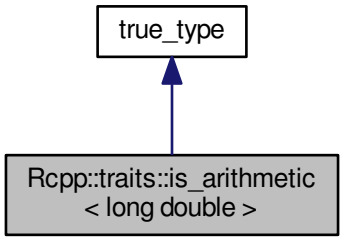
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

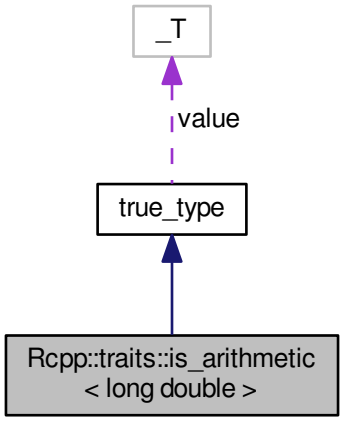
6.294 Rcpp::traits::is_arithmetic< long double > Struct Template Reference

`#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

6.294.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< long double >

Definition at line 97 of file is_arithmetic.h.

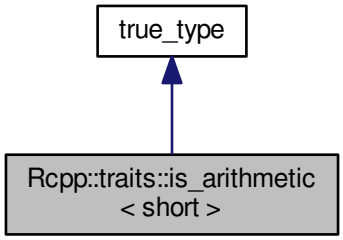
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

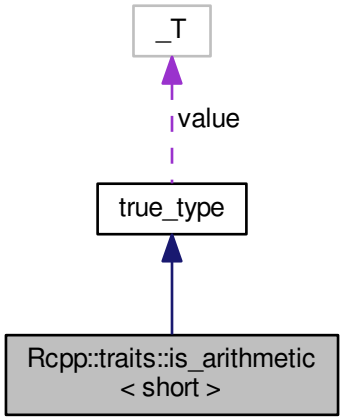
6.295 Rcpp::traits::is_arithmetic< short > Struct Template Reference

`#include <is_arithmetic.h>`

Inheritance diagram for Rcpp::traits::is_arithmetic< short >:



Collaboration diagram for Rcpp::traits::is_arithmetic< short >:



Additional Inherited Members

6.295.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< short >

Definition at line 33 of file is_arithmetic.h.

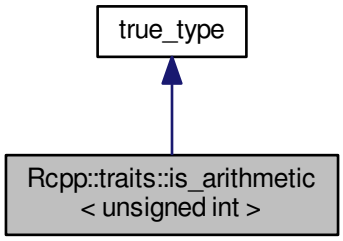
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

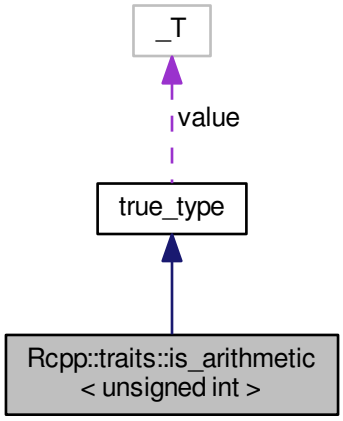
6.296 Rcpp::traits::is_arithmetic< unsigned int > Struct Template Reference

`#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

6.296.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< unsigned int >

Definition at line 51 of file is_arithmetic.h.

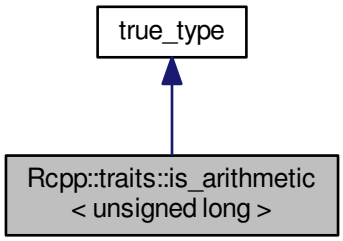
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

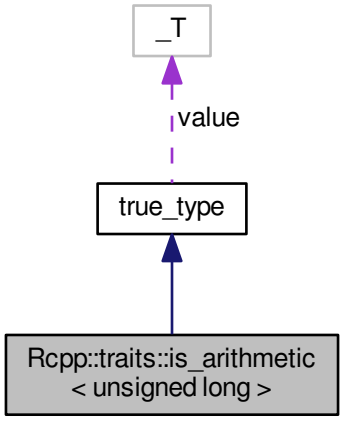
6.297 Rcpp::traits::is_arithmetic< unsigned long > Struct Template Reference

`#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

6.297.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< unsigned long >

Definition at line 63 of file is_arithmetic.h.

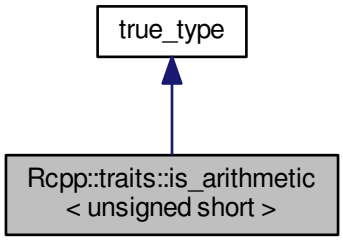
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

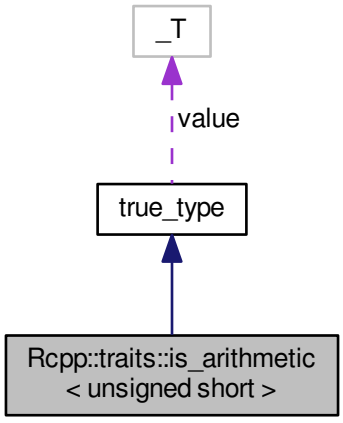
6.298 Rcpp::traits::is_arithmetic< unsigned short > Struct Template Reference

`#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

6.298.1 Detailed Description

template<>
struct Rcpp::traits::is_arithmetic< unsigned short >

Definition at line 39 of file is_arithmetic.h.

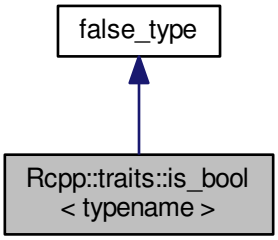
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

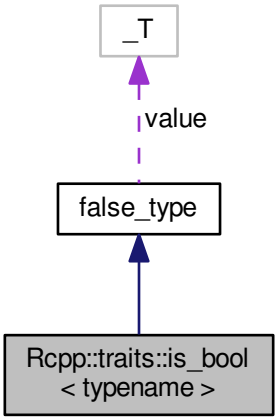
6.299 Rcpp::traits::is_bool< typename > Struct Template Reference

`#include <is_bool.h>`

Inheritance diagram for Rcpp::traits::is_bool< typename >:



Collaboration diagram for Rcpp::traits::is_bool< typename >:



Additional Inherited Members

6.299.1 Detailed Description

template<typename>
struct Rcpp::traits::is_bool< typename >

Definition at line 7 of file is_bool.h.

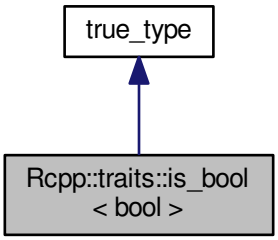
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

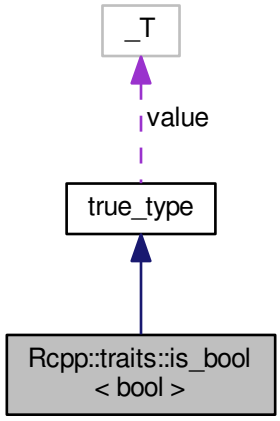
6.300 Rcpp::traits::is_bool< bool > Struct Template Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is_bool< bool >:



Collaboration diagram for Rcpp::traits::is_bool< bool >:



Additional Inherited Members

6.300.1 Detailed Description

template<>
struct Rcpp::traits::is_bool< bool >

Definition at line 11 of file is_bool.h.

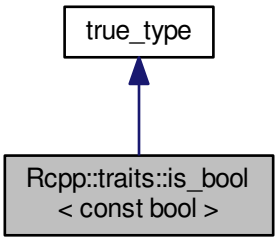
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

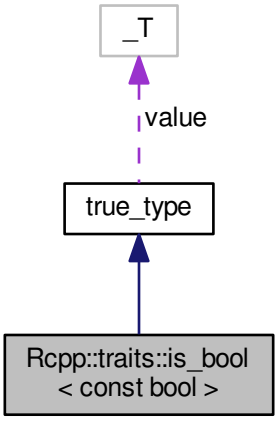
6.301 Rcpp::traits::is_bool< const bool > Struct Template Reference

`#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

6.301.1 Detailed Description

template<>
struct Rcpp::traits::is_bool< const bool >

Definition at line 15 of file is_bool.h.

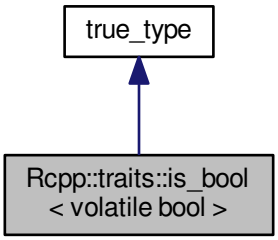
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

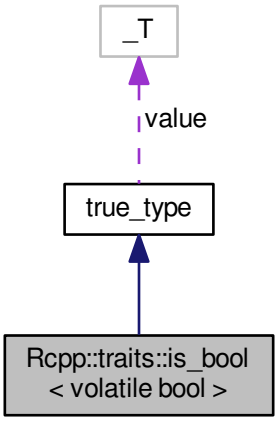
6.302 Rcpp::traits::is_bool< volatile bool > Struct Template Reference

`#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

6.302.1 Detailed Description

template<>
struct Rcpp::traits::is_bool< volatile bool >

Definition at line 19 of file is_bool.h.

The documentation for this struct was generated from the following file:

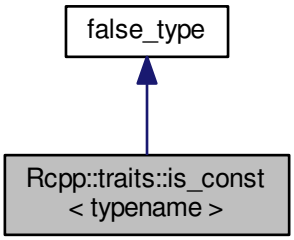
- inst/include/Rcpp/traits/is_bool.h

6.303 Rcpp::traits::is_const< typename > Struct Template Reference

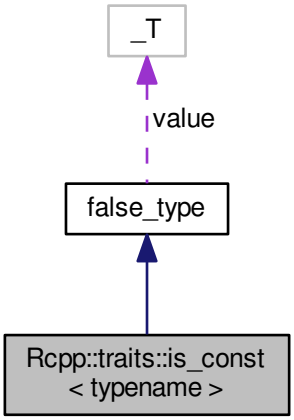
type properties [4.5.3].

`#include <is_const.h>`

Inheritance diagram for Rcpp::traits::is_const< typename >:



Collaboration diagram for Rcpp::traits::is_const< typename >:



Additional Inherited Members

6.303.1 Detailed Description

template<typename>
struct Rcpp::traits::is_const< typename >

type properties [4.5.3].

Definition at line 30 of file is_const.h.

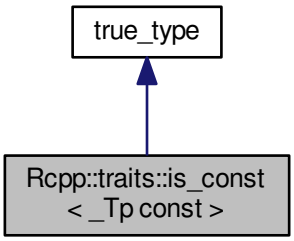
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_const.h

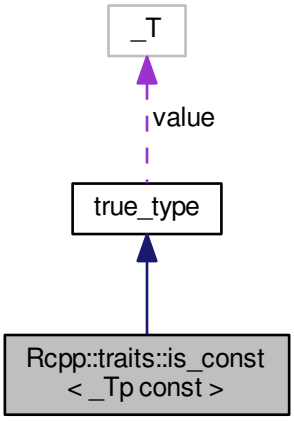
6.304 Rcpp::traits::is_const< _Tp const > Struct Template Reference

`#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

6.304.1 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

6.305 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 ()

6.305.1 Detailed Description

template<typename T, typename U>
class Rcpp::traits::is_convertible< T, U >

Definition at line 29 of file is_convertible.h.

6.305.2 Member Typedef Documentation

6.305.2.1 template<typename T , typename U > typedef char Rcpp::traits::is_convertible< T, U >::Small [private]

Definition at line 31 of file is_convertible.h.

6.305.2.2 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.

6.305.2.3 template<typename T , typename U > typedef bool Rcpp::traits::is_convertible< T, U >::value_type

Definition at line 39 of file is_convertible.h.

6.305.3 Member Function Documentation

6.305.3.1 template<typename T , typename U > static T Rcpp::traits::is_convertible< T, U >::MakeT ( ) [static], [private]

6.305.3.2 template<typename T , typename U > static Small Rcpp::traits::is_convertible< T, U >::Test (const U &) [static], [private]

6.305.3.3 template<typename T , typename U > static Big Rcpp::traits::is_convertible< T, U >::Test (...) [static], [private]

6.305.4 Member Data Documentation

6.305.4.1 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

6.306 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 ()

6.306.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >

Definition at line 185 of file tinyformat.h.

6.306.2 Member Function Documentation

6.306.2.1 template<typename T1 , typename T2 > static const T1& tinyformat::detail::is_convertible< T1, T2 >::makeT1 ( ) [static], [private]

6.306.2.2 template<typename T1 , typename T2 > static fail tinyformat::detail::is_convertible< T1, T2 >::tryConvert (...) [static], [private]

6.306.2.3 template<typename T1 , typename T2 > static succeed tinyformat::detail::is_convertible< T1, T2 >::tryConvert (const T2 &) [static], [private]

6.306.3 Member Data Documentation

6.306.3.1 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 206 of file tinyformat.h.

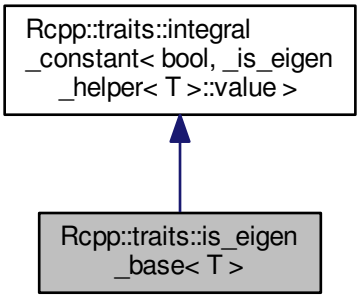
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

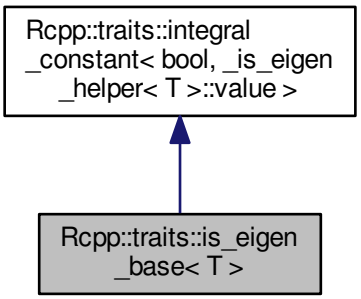
6.307 Rcpp::traits::is_eigen_base< T > Struct Template Reference

`#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

6.307.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_eigen_base< T >

Definition at line 42 of file is_eigen_base.h.

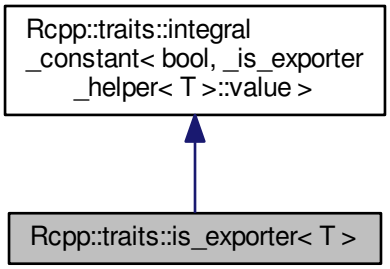
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_eigen_base.h

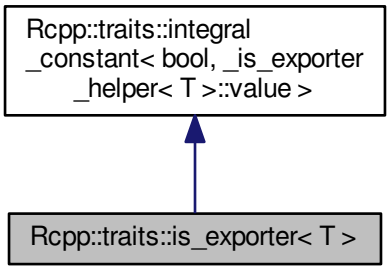
6.308 Rcpp::traits::is_exporter< T > Struct Template Reference

`#include <has_iterator.h>`

Inheritance diagram for Rcpp::traits::is_exporter< T >:



Collaboration diagram for Rcpp::traits::is_exporter< T >:



Additional Inherited Members

6.308.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_exporter< T >

Definition at line 115 of file has_iterator.h.

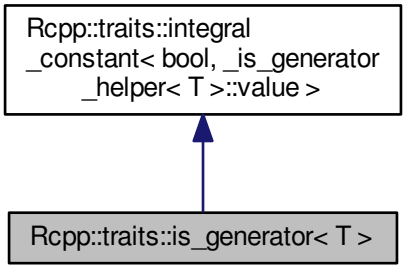
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

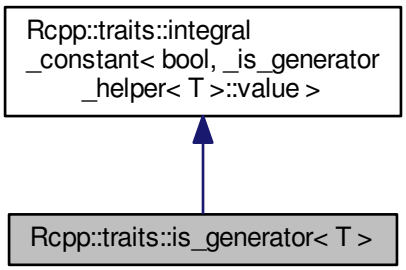
6.309 Rcpp::traits::is_generator< T > Struct Template Reference

`#include <has_iterator.h>`

Inheritance diagram for Rcpp::traits::is_generator< T >:



Collaboration diagram for Rcpp::traits::is_generator< T >:



Additional Inherited Members

6.309.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_generator< T >

Definition at line 118 of file has_iterator.h.

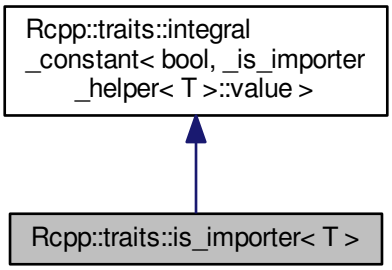
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

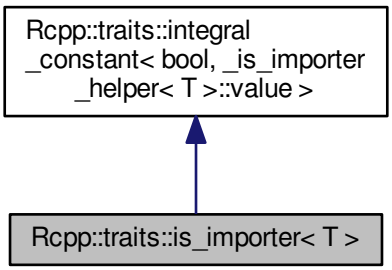
6.310 Rcpp::traits::is_importer< T > Struct Template Reference

`#include <has_iterator.h>`

Inheritance diagram for Rcpp::traits::is_importer< T >:



Collaboration diagram for Rcpp::traits::is_importer< T >:



Additional Inherited Members

6.310.1 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.

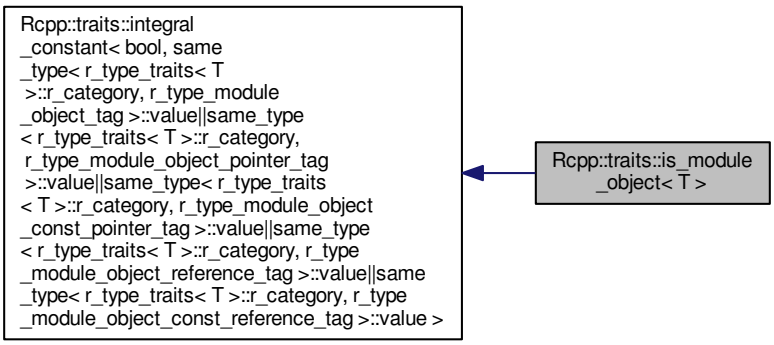
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

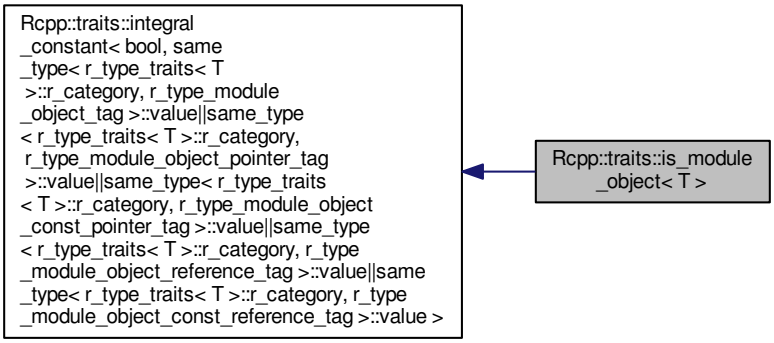
6.311 Rcpp::traits::is_module_object< T > Struct Template Reference

`#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

6.311.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_module_object< T >

Definition at line 29 of file is_module_object.h.

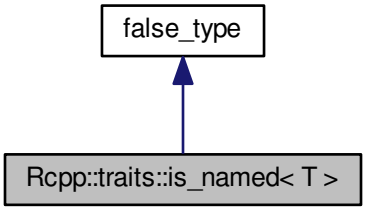
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_module_object.h

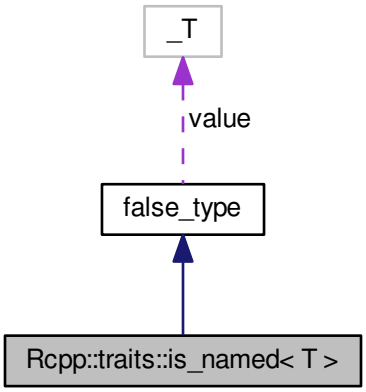
6.312 Rcpp::traits::is_named< T > Struct Template Reference

`#include <named_object.h>`

Inheritance diagram for Rcpp::traits::is_named< T >:



Collaboration diagram for Rcpp::traits::is_named< T >:



Additional Inherited Members

6.312.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_named< T >

Definition at line 62 of file named_object.h.

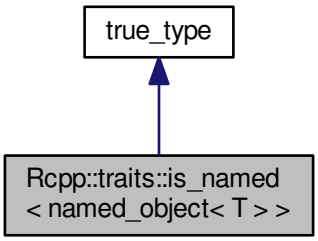
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

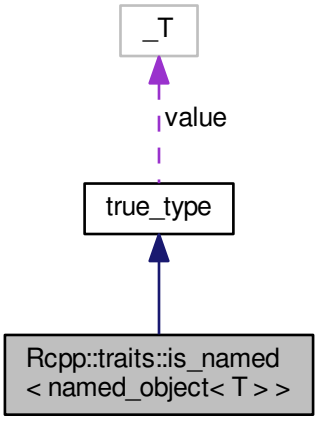
6.313 Rcpp::traits::is_named< named_object< T > > Struct Template Reference

`#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

6.313.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_named< named_object< T > >

Definition at line 63 of file named_object.h.

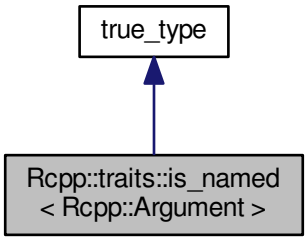
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

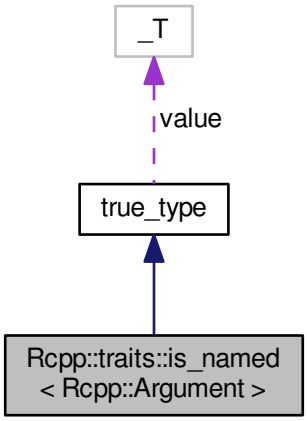
6.314 Rcpp::traits::is_named< Rcpp::Argument > Struct Template Reference

`#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

6.314.1 Detailed Description

template<>
struct Rcpp::traits::is_named< Rcpp::Argument >

Definition at line 64 of file named_object.h.

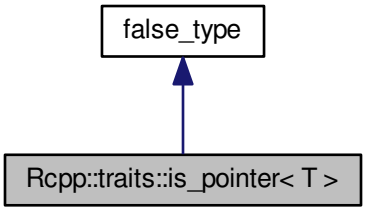
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

6.315 Rcpp::traits::is_pointer< T > Struct Template Reference

`#include <is_pointer.h>`

Inheritance diagram for Rcpp::traits::is_pointer< T >:



Additional Inherited Members

6.315.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_pointer< T >

Definition at line 29 of file is_pointer.h.

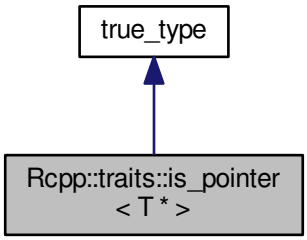
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_pointer.h

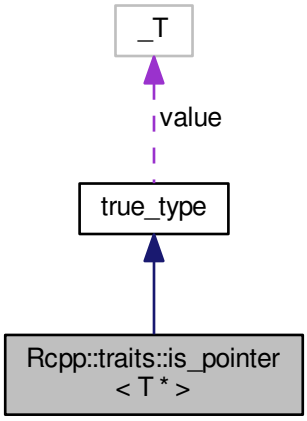
6.316 Rcpp::traits::is_pointer< T * > Struct Template Reference

`#include <is_pointer.h>`

Inheritance diagram for Rcpp::traits::is_pointer< T * >:



Collaboration diagram for Rcpp::traits::is_pointer< T * >:



Additional Inherited Members

6.316.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_pointer< T * >

Definition at line 30 of file is_pointer.h.

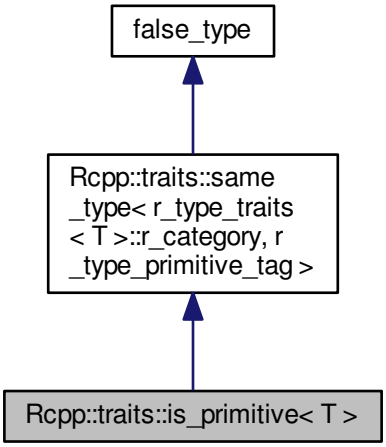
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_pointer.h

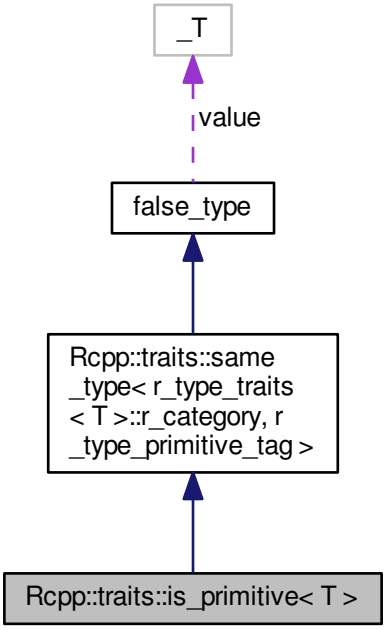
6.317 Rcpp::traits::is_primitive< T > Struct Template Reference

`#include <is_primitive.h>`

Inheritance diagram for Rcpp::traits::is_primitive< T >:



Collaboration diagram for Rcpp::traits::is_primitive< T >:



Additional Inherited Members

6.317.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_primitive< T >

Definition at line 29 of file is_primitive.h.

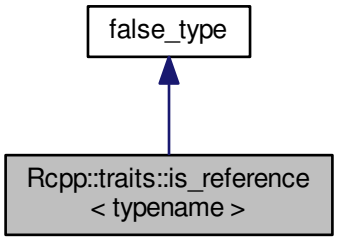
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_primitive.h

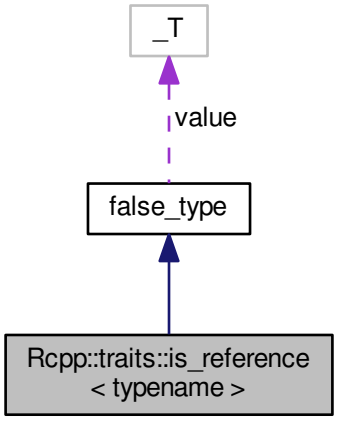
6.318 Rcpp::traits::is_reference< typename > Struct Template Reference

```
#include <is_reference.h>
```

Inheritance diagram for Rcpp::traits::is_reference< typename >:



Collaboration diagram for Rcpp::traits::is_reference< typename >:



Additional Inherited Members

6.318.1 Detailed Description

template<typename>
struct Rcpp::traits::is_reference< typename >

Definition at line 29 of file is_reference.h.

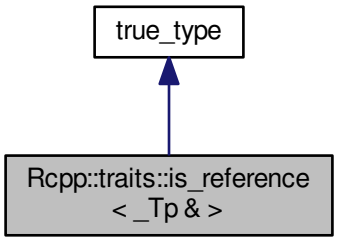
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_reference.h

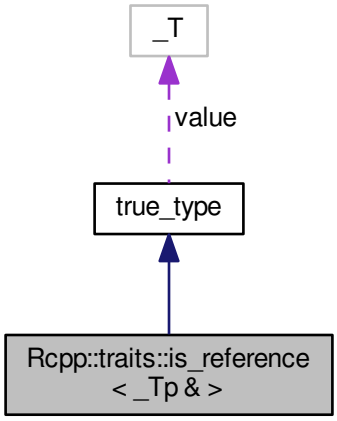
6.319 Rcpp::traits::is_reference< _Tp & > Struct Template Reference

`#include <is_reference.h>`

Inheritance diagram for Rcpp::traits::is_reference< _Tp & >:



Collaboration diagram for Rcpp::traits::is_reference< _Tp & >:



Additional Inherited Members

6.319.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::is_reference< _Tp & >

Definition at line 33 of file is_reference.h.

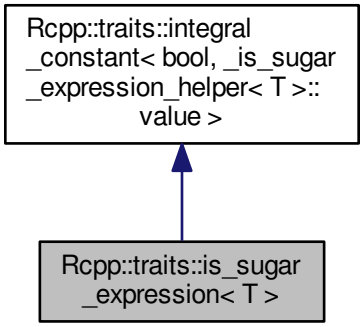
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_reference.h

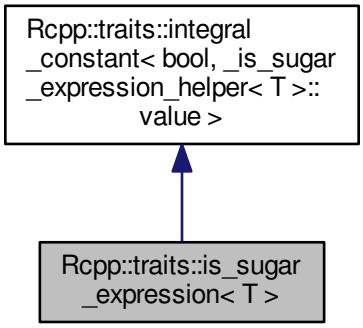
6.320 Rcpp::traits::is_sugar_expression< T > Struct Template Reference

`#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

6.320.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_sugar_expression< T >

Definition at line 42 of file is_sugar_expression.h.

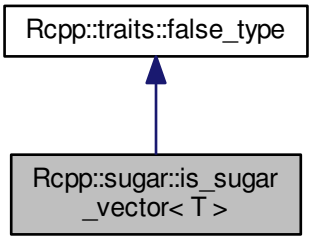
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_sugar_expression.h

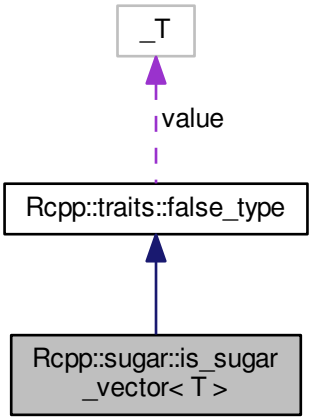
6.321 Rcpp::sugar::is_sugar_vector< T > Struct Template Reference

`#include <iterator.h>`

Inheritance diagram for Rcpp::sugar::is_sugar_vector< T >:



Collaboration diagram for Rcpp::sugar::is_sugar_vector< T >:



Additional Inherited Members

6.321.1 Detailed Description

template<typename T>
struct Rcpp::sugar::is_sugar_vector< T >

Definition at line 121 of file iterator.h.

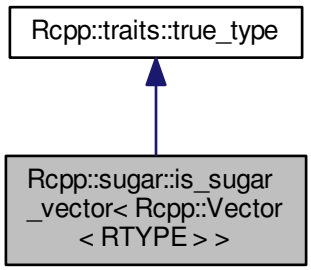
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

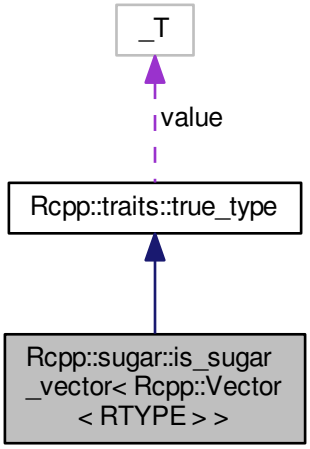
6.322 Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > > Struct Template Reference

`#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

6.322.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >

Definition at line 122 of file iterator.h.

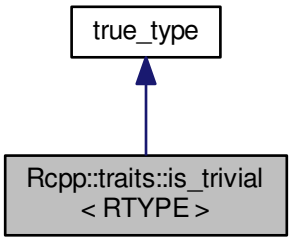
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

6.323 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

6.323.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::is_trivial< RTYPE >

Definition at line 28 of file is_trivial.h.

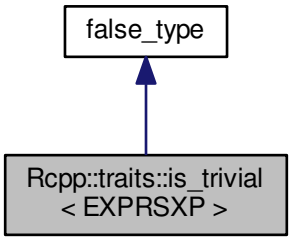
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_trivial.h

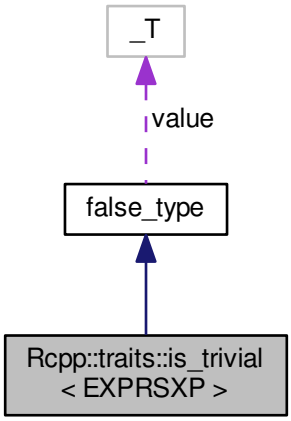
6.324 Rcpp::traits::is_trivial< EXPRSXP > Struct Template Reference

`#include <is_trivial.h>`

Inheritance diagram for Rcpp::traits::is_trivial< EXPRSXP >:



Collaboration diagram for Rcpp::traits::is_trivial< EXPRSXP >:



Additional Inherited Members

6.324.1 Detailed Description

template<>
struct Rcpp::traits::is_trivial< EXPRSXP >

Definition at line 30 of file is_trivial.h.

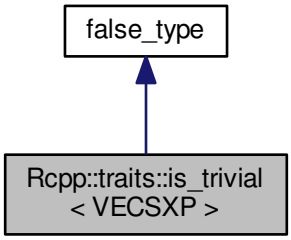
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_trivial.h

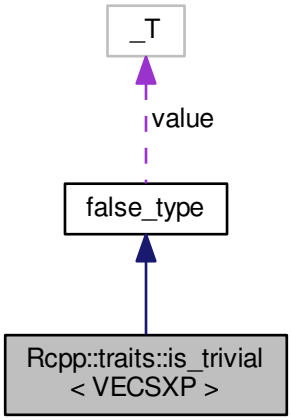
6.325 Rcpp::traits::is_trivial< VECSXP > Struct Template Reference

`#include <is_trivial.h>`

Inheritance diagram for Rcpp::traits::is_trivial< VECSXP >:



Collaboration diagram for Rcpp::traits::is_trivial< VECSXP >:



Additional Inherited Members

6.325.1 Detailed Description

template<>
struct Rcpp::traits::is_trivial< VECSXP >

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

6.326 tinyformat::detail::is_wchar< T > Struct Template Reference

`#include <tinyformat.h>`

Public Types

- typedef int tinyformat_wchar_is_not_supported

6.326.1 Detailed Description

template<typename T>
struct tinyformat::detail::is_wchar< T >

Definition at line 215 of file tinyformat.h.

6.326.2 Member Typedef Documentation

6.326.2.1 template<typename T> typedef int tinyformat::detail::is_wchar< T >::tinyformat_wchar_is_not_supported

Definition at line 215 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.327 tinyformat::detail::is_wchar< const wchar_t * > Struct Template Reference

`#include <tinyformat.h>`

6.327.1 Detailed Description

template<>
struct tinyformat::detail::is_wchar< const wchar_t * >

Definition at line 217 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.328 tinyformat::detail::is_wchar< const wchar_t[n]> Struct Template Reference

`#include <tinyformat.h>`

6.328.1 Detailed Description

template<int n>
struct tinyformat::detail::is_wchar< const wchar_t[n]>

Definition at line 218 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.329 tinyformat::detail::is_wchar< wchar_t * > Struct Template Reference

`#include <tinyformat.h>`

6.329.1 Detailed Description

template<>
struct tinyformat::detail::is_wchar< wchar_t * >

Definition at line 216 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.330 tinyformat::detail::is_wchar< wchar_t[n]> Struct Template Reference

`#include <tinyformat.h>`

6.330.1 Detailed Description

template<int n>
struct tinyformat::detail::is_wchar< wchar_t[n]>

Definition at line 219 of file tinyformat.h.

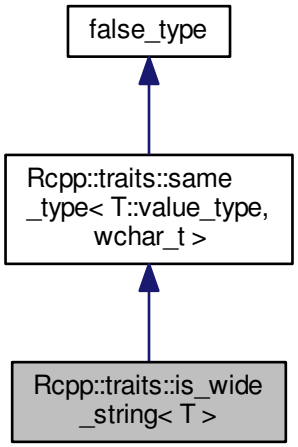
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

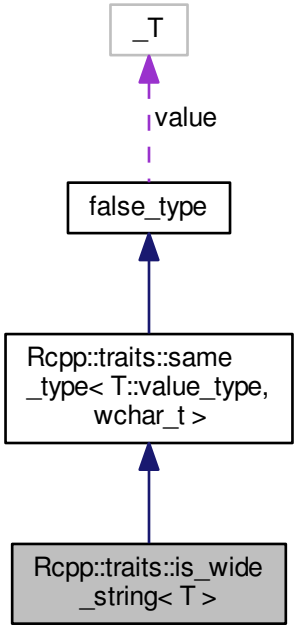
6.331 Rcpp::traits::is_wide_string< T > Struct Template Reference

`#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

6.331.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_wide_string< T >

Definition at line 30 of file is_wide_string.h.

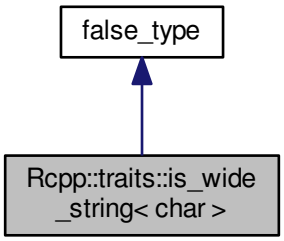
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

6.332 Rcpp::traits::is_wide_string< char > Struct Template 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

6.332.1 Detailed Description

template<>
struct Rcpp::traits::is_wide_string< char >

Definition at line 36 of file is_wide_string.h.

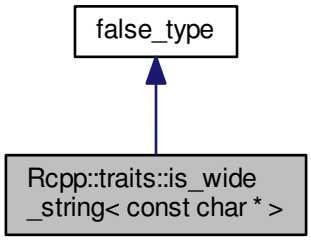
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

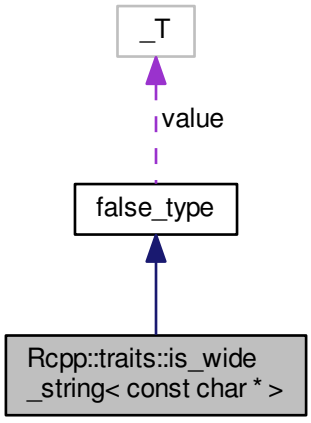
6.333 Rcpp::traits::is_wide_string< const char * > Struct Template Reference

`#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

6.333.1 Detailed Description

template<>
struct Rcpp::traits::is_wide_string< const char * >

Definition at line 33 of file is_wide_string.h.

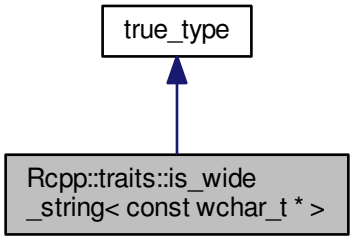
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

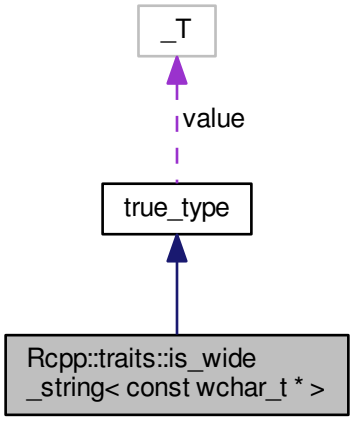
6.334 Rcpp::traits::is_wide_string< const wchar_t * > Struct Template Reference

`#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

6.334.1 Detailed Description

template<>
struct Rcpp::traits::is_wide_string< const wchar_t * >

Definition at line 32 of file is_wide_string.h.

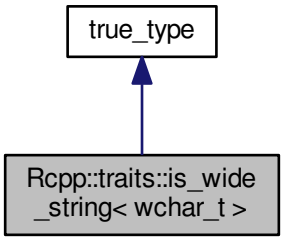
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

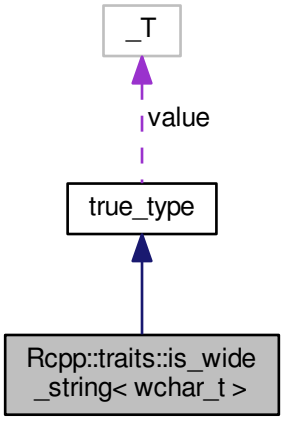
6.335 Rcpp::traits::is_wide_string< wchar_t > Struct Template Reference

`#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

6.335.1 Detailed Description

template<>
struct Rcpp::traits::is_wide_string< wchar_t >

Definition at line 35 of file is_wide_string.h.

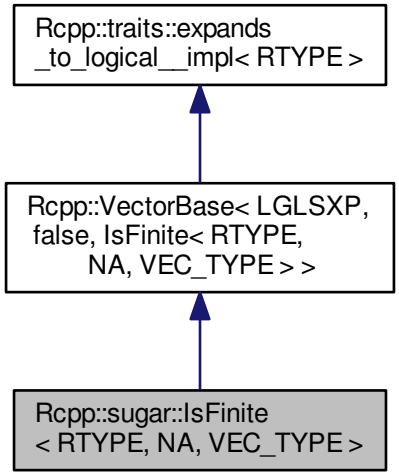
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

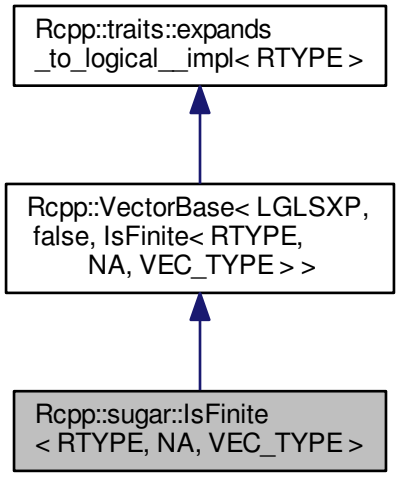
6.336 Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE > Class Template Reference

`#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

6.336.1 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.

6.336.2 Constructor & Destructor Documentation

6.336.2.1 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.

6.336.3 Member Function Documentation

6.336.3.1 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.

6.336.3.2 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.

6.336.4 Member Data Documentation

6.336.4.1 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().

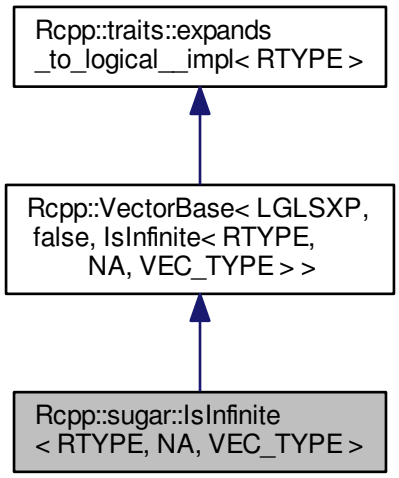
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_finite.h

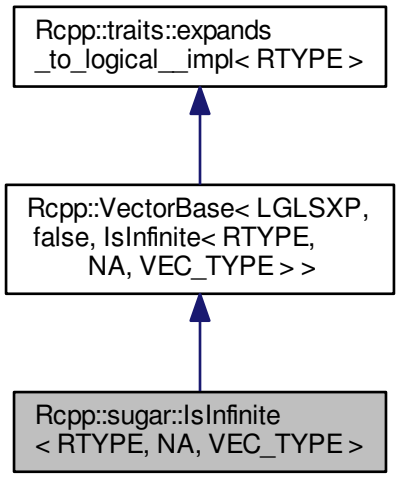
6.337 Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE > Class Template Reference

```
#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

6.337.1 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.

6.337.2 Constructor & Destructor Documentation

6.337.2.1 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.

6.337.3 Member Function Documentation

6.337.3.1 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.

6.337.3.2 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.

6.337.4 Member Data Documentation

6.337.4.1 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().

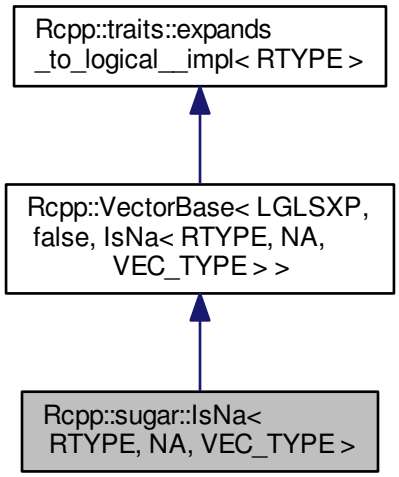
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_infinite.h

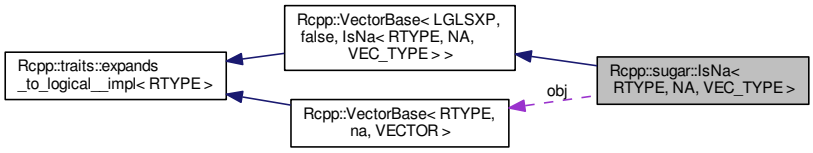
6.338 Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE > Class Template Reference

`#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

6.338.1 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.

6.338.2 Member Typedef Documentation

6.338.2.1 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.

6.338.2.2 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.

6.338.3 Constructor & Destructor Documentation

6.338.3.1 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.

6.338.4 Member Function Documentation

6.338.4.1 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.

6.338.4.2 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:

6.338.5 Member Data Documentation

6.338.5.1 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().

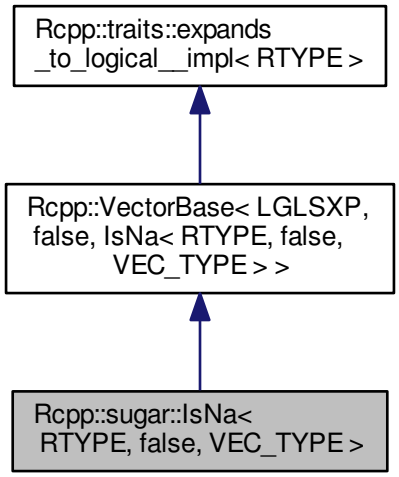
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

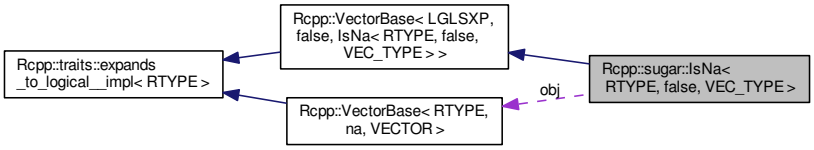
6.339 Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE > Class Template Reference

```
#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

6.339.1 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.

6.339.2 Member Typedef Documentation

6.339.2.1 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.

6.339.2.2 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.

6.339.3 Constructor & Destructor Documentation

6.339.3.1 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.

6.339.4 Member Function Documentation

6.339.4.1 template<int RTYPE, typename VEC_TYPE > int Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::operator[ ] (R_xlen_t *i*) const [inline]

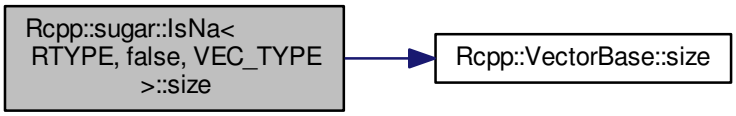
Definition at line 58 of file is_na.h.

6.339.4.2 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:



6.339.5 Member Data Documentation

6.339.5.1 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.

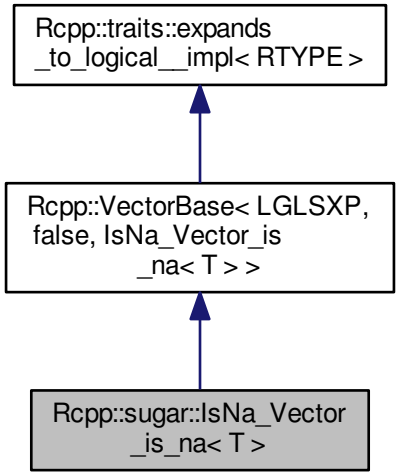
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

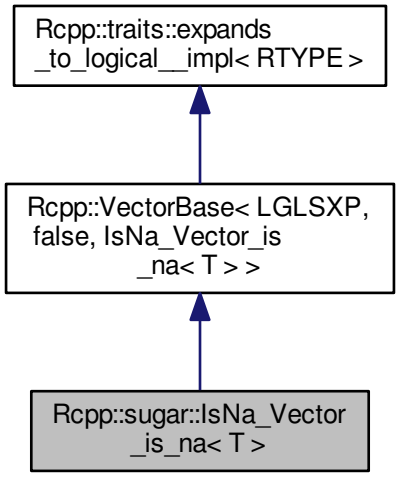
6.340 Rcpp::sugar::IsNa_Vector_is_na< T > Class Template Reference

```
#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

6.340.1 Detailed Description

template<typename T>
class Rcpp::sugar::IsNa_Vector_is_na< T >

Definition at line 70 of file is_na.h.

6.340.2 Constructor & Destructor Documentation

6.340.2.1 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.

6.340.3 Member Function Documentation

6.340.3.1 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.

6.340.3.2 template<typename T > R_xlen_t Rcpp::sugar::IsNa_Vector_is_na< T >::size ( ) const [inline]

Definition at line 78 of file is_na.h.

6.340.4 Member Data Documentation

6.340.4.1 template<typename T > const T& Rcpp::sugar::IsNa_Vector_is_na< T >::ref [private]

Definition at line 81 of file is_na.h.

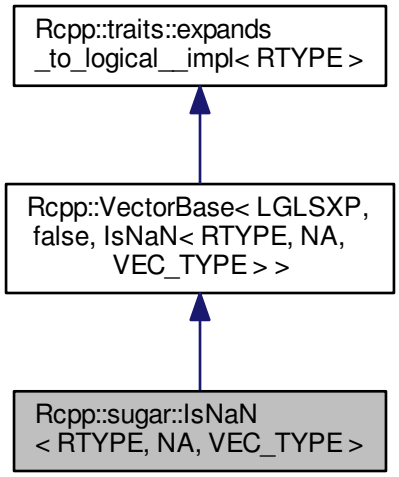
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

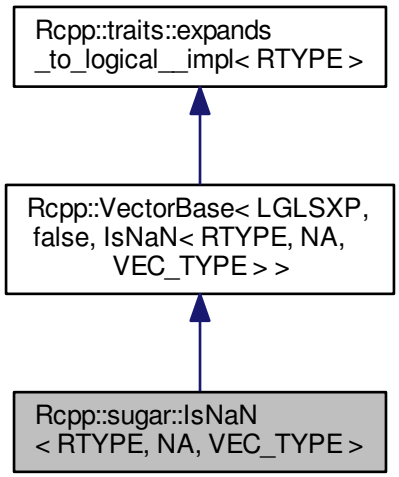
6.341 Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE > Class Template Reference

`#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

6.341.1 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.

6.341.2 Constructor & Destructor Documentation

6.341.2.1 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.

6.341.3 Member Function Documentation

6.341.3.1 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.

6.341.3.2 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.

6.341.4 Member Data Documentation

6.341.4.1 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().

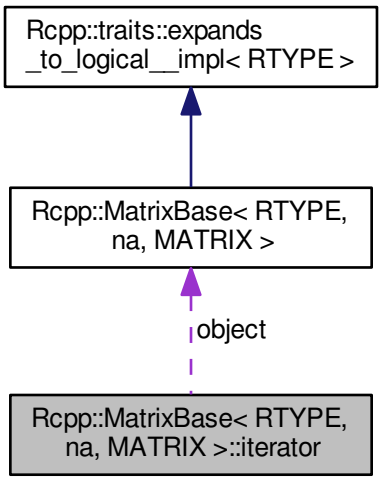
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_nan.h

6.342 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

6.342.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
class Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator

Definition at line 50 of file MatrixBase.h.

6.342.2 Member Typedef Documentation

6.342.2.1 template<int RTYPE, bool na, typename MATRIX> typedef R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::difference_type

Definition at line 54 of file MatrixBase.h.

6.342.2.2 template<int RTYPE, bool na, typename MATRIX> typedef std::random_access_iterator_tag Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator_category

Definition at line 56 of file MatrixBase.h.

6.342.2.3 template<int RTYPE, bool na, typename MATRIX> typedef stored_type* Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::pointer

Definition at line 53 of file MatrixBase.h.

6.342.2.4 template<int RTYPE, bool na, typename MATRIX> typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::reference

Definition at line 52 of file MatrixBase.h.

6.342.2.5 template<int RTYPE, bool na, typename MATRIX> typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::value_type

Definition at line 55 of file MatrixBase.h.

6.342.3 Constructor & Destructor Documentation

6.342.3.1 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 58 of file MatrixBase.h.

6.342.3.2 template<int RTYPE, bool na, typename MATRIX> Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator (const iterator & *other*) [inline]

Definition at line 64 of file MatrixBase.h.

6.342.4 Member Function Documentation

6.342.4.1 template<int RTYPE, bool na, typename MATRIX> R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index ( ) const [inline], [private]

Definition at line 152 of file MatrixBase.h.

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>=().

6.342.4.2 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!= (const iterator & *y*) const [inline]

Definition at line 121 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j.

6.342.4.3 template<int RTYPE, bool na, typename MATRIX> reference Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator* ( ) [inline]

Definition at line 111 of file MatrixBase.h.

6.342.4.4 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+ (difference_type *n*) const [inline]

Definition at line 95 of file MatrixBase.h.

6.342.4.5 template<int RTYPE, bool na, typename MATRIX> iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ ( ) [inline]

Definition at line 67 of file MatrixBase.h.

6.342.4.6 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ (int) [inline]

Definition at line 75 of file MatrixBase.h.

6.342.4.7 template<int RTYPE, bool na, typename MATRIX> iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+= (difference_type *n*) [inline]

Definition at line 102 of file MatrixBase.h.

6.342.4.8 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (difference_type *n*) const [inline]

Definition at line 98 of file MatrixBase.h.

6.342.4.9 template<int RTYPE, bool na, typename MATRIX> difference_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (const iterator & *other*) const [inline]

Definition at line 137 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

6.342.4.10 template<int RTYPE, bool na, typename MATRIX> iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- ( ) [inline]

Definition at line 81 of file MatrixBase.h.

6.342.4.11 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- (int) [inline]

Definition at line 89 of file MatrixBase.h.

6.342.4.12 template<int RTYPE, bool na, typename MATRIX> iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-= (difference_type *n*) [inline]

Definition at line 106 of file MatrixBase.h.

6.342.4.13 template<int RTYPE, bool na, typename MATRIX> pointer Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-> ( ) [inline]

Definition at line 114 of file MatrixBase.h.

6.342.4.14 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator< (const iterator & *other*) const [inline]

Definition at line 124 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

6.342.4.15 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<= (const iterator & *other*) const [inline]

Definition at line 130 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

6.342.4.16 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator== (const iterator & *y*) const [inline]

Definition at line 118 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j.

6.342.4.17 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator> (const iterator & *other*) const [inline]

Definition at line 127 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

6.342.4.18 template<int RTYPE, bool na, typename MATRIX> bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>= (const iterator & *other*) const [inline]

Definition at line 133 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

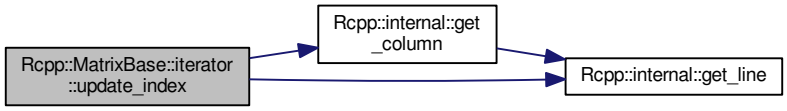
Here is the call graph for this function:

6.342.4.19 template<int RTYPE, bool na, typename MATRIX> void Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index (int *index_*) [inline], [private]

Definition at line 147 of file MatrixBase.h.

References Rcpp::internal::get_column(), and Rcpp::internal::get_line().

Here is the call graph for this function:



6.342.5 Member Data Documentation

6.342.5.1 template<int RTYPE, bool na, typename MATRIX> int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i [private]

Definition at line 144 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!=(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==().

6.342.5.2 template<int RTYPE, bool na, typename MATRIX> int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j [private]

Definition at line 144 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!=(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==().

6.342.5.3 template<int RTYPE, bool na, typename MATRIX> int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nc [private]

Definition at line 145 of file MatrixBase.h.

6.342.5.4 template<int RTYPE, bool na, typename MATRIX> int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr [private]

Definition at line 145 of file MatrixBase.h.

6.342.5.5 template<int RTYPE, bool na, typename MATRIX> const MatrixBase& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object [private]

Definition at line 143 of file MatrixBase.h.

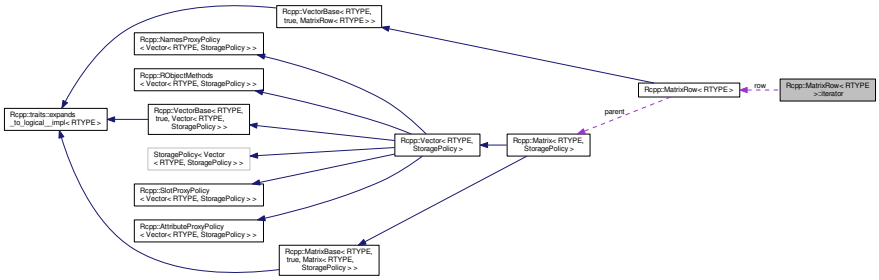
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

6.343 Rcpp::MatrixRow< RTYPE >::iterator Class Reference

`#include <MatrixRow.h>`

Collaboration diagram for Rcpp::MatrixRow< RTYPE >::iterator:



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 traits::r_vector_proxy< RTYPE >::type reference
- typedef std::iterator_traits< vector_iterator >::pointer pointer
- typedef std::random_access_iterator_tag iterator_category

Public Member Functions

- iterator (const iterator &other)
- iterator (MatrixRow &row_, int index_)
- iterator & operator++ ()
- iterator operator++ (int)
- iterator & operator-- ()
- iterator operator-- (int)
- iterator operator+ (difference_type n) const
- iterator operator- (difference_type n) const
- difference_type operator- (const iterator &other) const
- iterator & operator+= (difference_type n)
- iterator & operator-= (difference_type n)
- reference operator* ()
- pointer operator-> ()
- bool operator== (const iterator &other)
- bool operator!= (const iterator &other)
- bool operator< (const iterator &other)
- bool operator> (const iterator &other)
- bool operator<= (const iterator &other)
- bool operator>= (const iterator &other)
- reference operator[ ] (int i) const
- difference_type operator- (const iterator &other)

Private Attributes

- MatrixRow & row
- int index

6.343.1 Detailed Description

template<int RTYPE>
class Rcpp::MatrixRow< RTYPE >::iterator

Definition at line 36 of file MatrixRow.h.

6.343.2 Member Typedef Documentation

6.343.2.1 template<int RTYPE> typedef int Rcpp::MatrixRow< RTYPE >::iterator::difference_type

Definition at line 40 of file MatrixRow.h.

6.343.2.2 template<int RTYPE> typedef std::random_access_iterator_tag Rcpp::MatrixRow< RTYPE >::iterator::iterator_category

Definition at line 45 of file MatrixRow.h.

6.343.2.3 template<int RTYPE> typedef std::iterator_traits<vector_iterator>::pointer Rcpp::MatrixRow< RTYPE >::iterator::pointer

Definition at line 43 of file MatrixRow.h.

6.343.2.4 template<int RTYPE> typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iterator::reference

Definition at line 42 of file MatrixRow.h.

6.343.2.5 template<int RTYPE> typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iterator::value_type

Definition at line 41 of file MatrixRow.h.

6.343.2.6 template<int RTYPE> typedef traits::r_vector_iterator<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iterator::vector_iterator

Definition at line 38 of file MatrixRow.h.

6.343.3 Constructor & Destructor Documentation

6.343.3.1 template<int RTYPE> Rcpp::MatrixRow< RTYPE >::iterator::iterator (const iterator & *other*) [inline]

Definition at line 47 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::begin(), Rcpp::MatrixRow< RTYPE >::end(), Rcpp::MatrixRow< RTY↩PE >::iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+(), Rcpp::MatrixRow< RTY↩PE >::iterator::operator-(), and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-().

6.343.3.2 template<int RTYPE> Rcpp::MatrixRow< RTYPE >::iterator::iterator (MatrixRow & *row_*, int *index_*) [inline]

Definition at line 48 of file MatrixRow.h.

6.343.4 Member Function Documentation

6.343.4.1 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator!= (const iterator & *other*) [inline]

Definition at line 85 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.2 template<int RTYPE> reference Rcpp::MatrixRow< RTYPE >::iterator::operator* ( ) [inline]

Definition at line 77 of file MatrixRow.h.

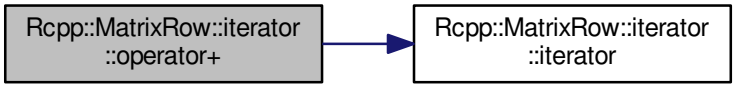
References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.343.4.3 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::iterator::operator+ (difference_type *n*) const [inline]

Definition at line 70 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp↩::MatrixRow< RTYPE >::iterator::row.

Here is the call graph for this function:



6.343.4.4 template<int RTYPE> iterator& Rcpp::MatrixRow< RTYPE >::iterator::operator++ ( ) [inline]

Definition at line 50 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.5 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::iterator::operator++ (int) [inline]

Definition at line 54 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.6 template<int RTYPE> iterator& Rcpp::MatrixRow< RTYPE >::iterator::operator+= (difference_type *n*) [inline]

Definition at line 74 of file MatrixRow.h.

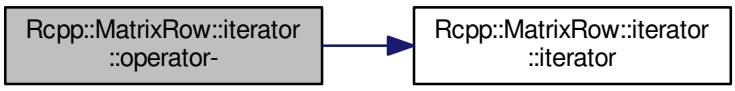
References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.7 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::iterator::operator- (difference_type *n*) const [inline]

Definition at line 71 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp↩ ::MatrixRow< RTYPE >::iterator::row.

Here is the call graph for this function:



6.343.4.8 template<int RTYPE> difference_type Rcpp::MatrixRow< RTYPE >::iterator::operator- (const iterator & *other*) const [inline]

Definition at line 72 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.9 template<int RTYPE> difference_type Rcpp::MatrixRow< RTYPE >::iterator::operator- (const iterator & *other*) [inline]

Definition at line 95 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.10 template<int RTYPE> iterator& Rcpp::MatrixRow< RTYPE >::iterator::operator-- ( ) [inline]

Definition at line 60 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.11 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::iterator::operator-- (int) [inline]

Definition at line 64 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.12 template<int RTYPE> iterator& Rcpp::MatrixRow< RTYPE >::iterator::operator-= (difference_type *n*) [inline]

Definition at line 75 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.13 template<int RTYPE> pointer Rcpp::MatrixRow< RTYPE >::iterator::operator-> ( ) [inline]

Definition at line 80 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.343.4.14 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator< (const iterator & *other*) [inline]

Definition at line 86 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.15 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator<= (const iterator & *other*) [inline]

Definition at line 88 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.16 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator== (const iterator & *other*) [inline]

Definition at line 84 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.17 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator> (const iterator & *other*) [inline]

Definition at line 87 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.18 template<int RTYPE> bool Rcpp::MatrixRow< RTYPE >::iterator::operator>= (const iterator & *other*) [inline]

Definition at line 89 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index.

6.343.4.19 template<int RTYPE> reference Rcpp::MatrixRow< RTYPE >::iterator::operator[ ] (int *i*) const [inline]

Definition at line 91 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::index, and Rcpp::MatrixRow< RTYPE >::iterator::row.

6.343.5 Member Data Documentation

6.343.5.1 template<int RTYPE> int Rcpp::MatrixRow< RTYPE >::iterator::index [private]

Definition at line 101 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iterator::operator!=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator!=(), Rcpp::MatrixRow< RTYPE >::iterator::operator*(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator*(), Rcpp::MatrixRow< RTYPE >::iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+(), Rcpp::MatrixRow< RTYPE >::iterator::operator++(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++(), Rcpp::MatrixRow< RTYPE >::iterator::operator+=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+=(), Rcpp::MatrixRow< RTYPE >::iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-(), Rcpp::MatrixRow< RTYPE >::iterator::operator--(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator--(), Rcpp::MatrixRow< RTYPE >::iterator::operator-=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-=(), Rcpp::MatrixRow< RTYPE >::iterator::operator->(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator->(), Rcpp::MatrixRow< RTYPE >::iterator::operator<(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<(), Rcpp::MatrixRow< RTYPE >::iterator::operator<=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<=(), Rcpp::MatrixRow< RTYPE >::iterator::operator==(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator==(), Rcpp::MatrixRow< RTYPE >::iterator::operator>(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>(), Rcpp::MatrixRow< RTYPE >::iterator::operator>=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>=(), Rcpp::MatrixRow< RTYPE >::iterator::operator[ ](), and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[ ]().

6.343.5.2 template<int RTYPE> MatrixRow& Rcpp::MatrixRow< RTYPE >::iterator::row [private]

Definition at line 100 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iterator::operator*(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator*(), Rcpp::MatrixRow< RTYPE >::iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+(), Rcpp::MatrixRow< RTYPE >::iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-(), Rcpp::MatrixRow< RTYPE >::iterator::operator->(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator->(), Rcpp::MatrixRow< RTYPE >::iterator::operator[](), Rcpp::MatrixRow< RTYPE >::operator[](), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[](), and Rcpp::ConstMatrixRow< RTYPE >::operator[]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.344 Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator Class 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

Public Member Functions

- iterator (const VectorBase &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[] (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 VECTOR & object
- R_xlen_t index

6.344.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
class Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator

Definition at line 51 of file VectorBase.h.

6.344.2 Member Typedef Documentation

6.344.2.1 template<int RTYPE, bool na, typename VECTOR> typedef R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::difference_type

Definition at line 55 of file VectorBase.h.

6.344.2.2 template<int RTYPE, bool na, typename VECTOR> typedef std::random_access_iterator_tag Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::iterator_category

Definition at line 57 of file VectorBase.h.

6.344.2.3 template<int RTYPE, bool na, typename VECTOR> typedef stored_type* Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::pointer

Definition at line 54 of file VectorBase.h.

6.344.2.4 template<int RTYPE, bool na, typename VECTOR> typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::reference

Definition at line 53 of file VectorBase.h.

6.344.2.5 template<int RTYPE, bool na, typename VECTOR> typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::value_type

Definition at line 56 of file VectorBase.h.

6.344.3 Constructor & Destructor Documentation

6.344.3.1 template<int RTYPE, bool na, typename VECTOR> Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::iterator (const VectorBase & *object_*, R_xlen_t *index_*) [inline]

Definition at line 59 of file VectorBase.h.

6.344.3.2 template<int RTYPE, bool na, typename VECTOR> Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::iterator (const iterator & *other*) [inline]

Definition at line 60 of file VectorBase.h.

6.344.4 Member Function Documentation

6.344.4.1 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator!= (const iterator & *y*) const [inline]

Definition at line 112 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.2 template<int RTYPE, bool na, typename VECTOR> reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator* ( ) [inline]

Definition at line 102 of file VectorBase.h.

6.344.4.3 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator+ (difference_type *n*) const [inline]

Definition at line 82 of file VectorBase.h.

6.344.4.4 template<int RTYPE, bool na, typename VECTOR> iterator& Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator++ ( ) [inline]

Definition at line 62 of file VectorBase.h.

6.344.4.5 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator++ (int) [inline]

Definition at line 66 of file VectorBase.h.

6.344.4.6 template<int RTYPE, bool na, typename VECTOR> iterator& Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator+= (difference_type *n*) [inline]

Definition at line 89 of file VectorBase.h.

6.344.4.7 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator- (difference_type *n*) const [inline]

Definition at line 85 of file VectorBase.h.

6.344.4.8 template<int RTYPE, bool na, typename VECTOR> difference_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator- (const iterator & *other*) const [inline]

Definition at line 128 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.9 template<int RTYPE, bool na, typename VECTOR> iterator& Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator-- ( ) [inline]

Definition at line 72 of file VectorBase.h.

6.344.4.10 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator-- (int) [inline]

Definition at line 76 of file VectorBase.h.

6.344.4.11 template<int RTYPE, bool na, typename VECTOR> iterator& Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator-= (difference_type *n*) [inline]

Definition at line 93 of file VectorBase.h.

6.344.4.12 template<int RTYPE, bool na, typename VECTOR> pointer Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator-> ( ) [inline]

Definition at line 105 of file VectorBase.h.

6.344.4.13 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator< (const iterator & *other*) const [inline]

Definition at line 115 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.14 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator<= (const iterator & *other*) const [inline]

Definition at line 121 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.15 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator== (const iterator & *y*) const [inline]

Definition at line 109 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.16 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator> (const iterator & *other*) const [inline]

Definition at line 118 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.17 template<int RTYPE, bool na, typename VECTOR> bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator>= (const iterator & *other*) const [inline]

Definition at line 124 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index.

6.344.4.18 template<int RTYPE, bool na, typename VECTOR> reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator[] (R_xlen_t *i*) [inline]

Definition at line 98 of file VectorBase.h.

6.344.5 Member Data Documentation

6.344.5.1 template<int RTYPE, bool na, typename VECTOR> R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::index [private]

Definition at line 135 of file VectorBase.h.

Referenced by Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator!=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator-(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator<(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator<=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator==(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator>(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::operator>=().

6.344.5.2 template<int RTYPE, bool na, typename VECTOR> const VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator::object [private]

Definition at line 134 of file VectorBase.h.

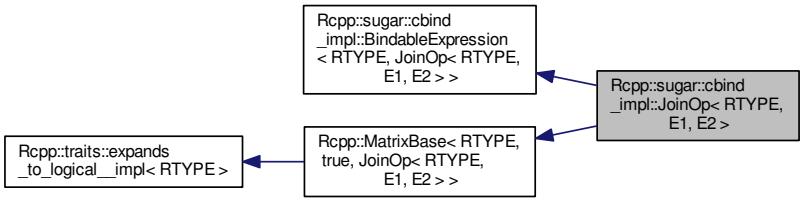
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

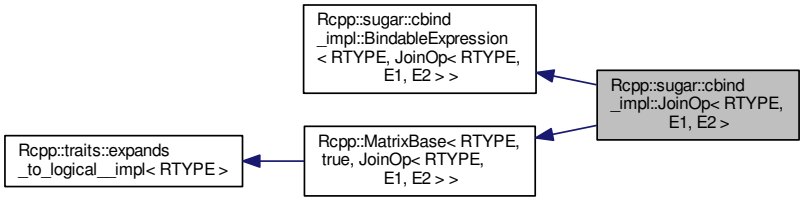
6.345 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 > Class Template Reference

`#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

6.345.1 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.

6.345.2 Member Typedef Documentation

6.345.2.1 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.

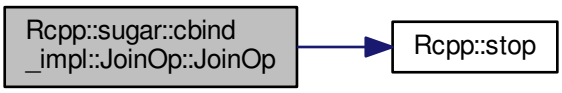
6.345.3 Constructor & Destructor Documentation

6.345.3.1 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::stop().

Here is the call graph for this function:



6.345.4 Member Function Documentation

6.345.4.1 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.

6.345.4.2 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.

6.345.4.3 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.

6.345.4.4 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.

6.345.4.5 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.

6.345.5 Member Data Documentation

6.345.5.1 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.

6.345.5.2 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.

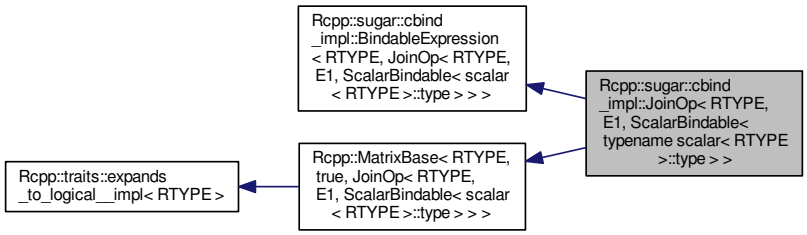
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

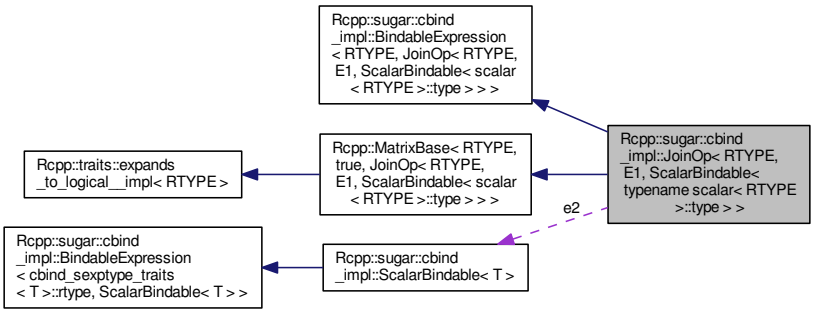
6.346 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

`#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

6.346.1 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.

6.346.2 Member Typedef Documentation

6.346.2.1 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.

6.346.2.2 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.

6.346.3 Constructor & Destructor Documentation

6.346.3.1 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.

6.346.4 Member Function Documentation

6.346.4.1 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.

6.346.4.2 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.

6.346.4.3 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.

6.346.4.4 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.

6.346.4.5 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.

6.346.5 Member Data Documentation

6.346.5.1 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.

6.346.5.2 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.

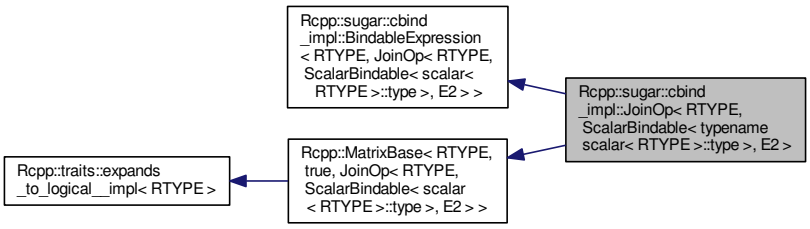
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

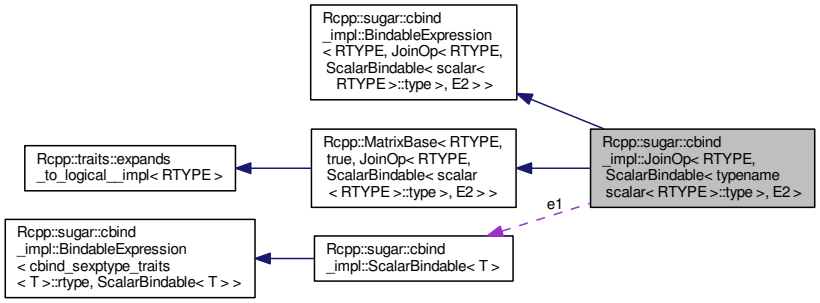
6.347 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > Class Template Reference

`#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

6.347.1 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.

6.347.2 Member Typedef Documentation

6.347.2.1 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.

6.347.2.2 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.

6.347.3 Constructor & Destructor Documentation

6.347.3.1 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.

6.347.4 Member Function Documentation

6.347.4.1 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.

6.347.4.2 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.

6.347.4.3 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.

6.347.4.4 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.

6.347.4.5 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.

6.347.5 Member Data Documentation

6.347.5.1 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.

6.347.5.2 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.

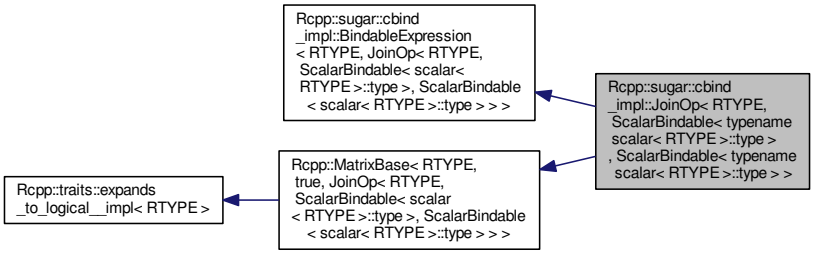
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

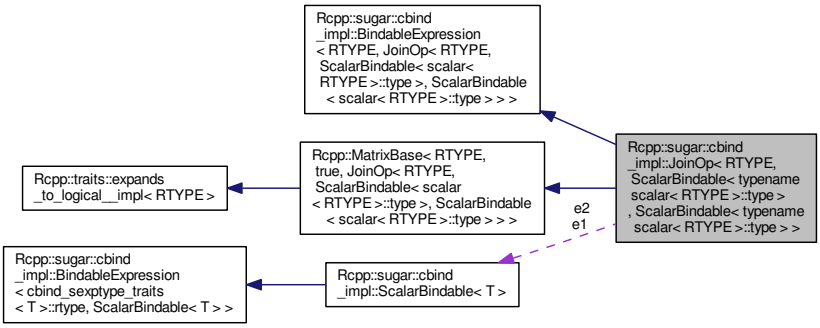
6.348 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

`#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

6.348.1 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.

6.348.2 Member Typedef Documentation

6.348.2.1 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.

6.348.2.2 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.

6.348.2.3 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.

6.348.3 Constructor & Destructor Documentation

6.348.3.1 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.

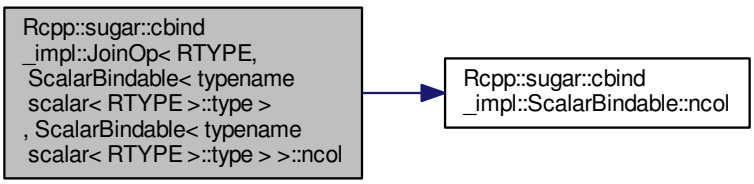
6.348.4 Member Function Documentation

6.348.4.1 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::ScalarBindable< T >::ncol().

Here is the call graph for this function:

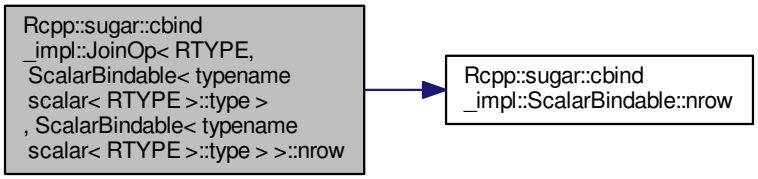


6.348.4.2 template<int RTYPE> R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::nrow ( ) const [inline]

Definition at line 314 of file cbind.h.

References Rcpp::sugar::cbind_impl::ScalarBindable< T >::nrow().

Here is the call graph for this function:



6.348.4.3 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]

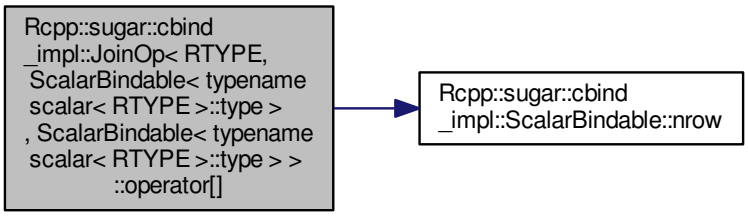
Definition at line 322 of file cbind.h.

6.348.4.4 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::ScalarBindable< T >::nrow().

Here is the call graph for this function:

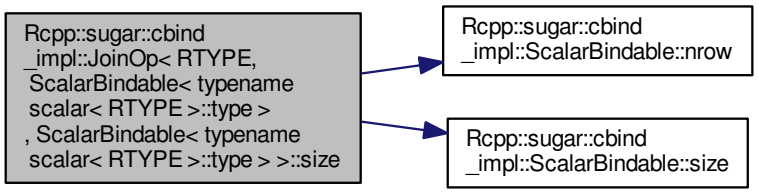


6.348.4.5 template<int RTYPE> R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::size ( ) const [inline]

Definition at line 312 of file cbind.h.

References Rcpp::sugar::cbind_impl::ScalarBindable< T >::nrow(), and Rcpp::sugar::cbind_impl::ScalarBindable< T >::size().

Here is the call graph for this function:



6.348.5 Member Data Documentation

6.348.5.1 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.

6.348.5.2 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.

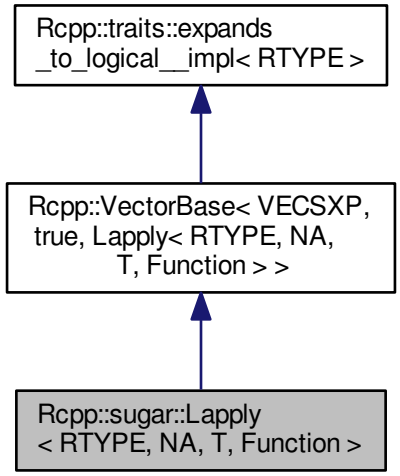
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

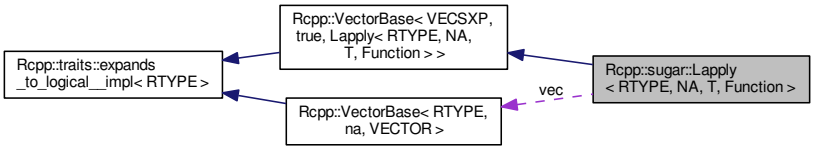
6.349 Rcpp::sugar::Lapply< RTYPE, NA, T, Function > Class Template Reference

`#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

6.349.1 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.

6.349.2 Member Typedef Documentation

6.349.2.1 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.

6.349.2.2 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.

6.349.3 Constructor & Destructor Documentation

6.349.3.1 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.

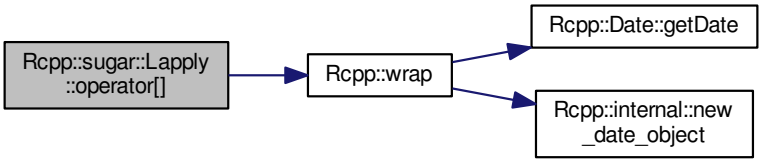
6.349.4 Member Function Documentation

6.349.4.1 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:

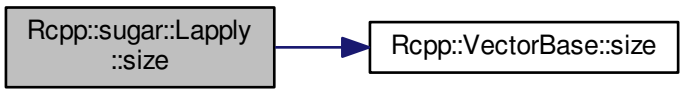


6.349.4.2 template<int RTYPE, bool NA, typename T, typename Function> R_xlen_t Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size ( ) const [inline]

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:



6.349.5 Member Data Documentation

6.349.5.1 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::lapply(), and Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[ ]().

6.349.5.2 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

6.350 Rcpp::sugar::Lazy< T, EXPR > Class Template Reference

`#include <Lazy.h>`

Public Member Functions

- operator T () const

6.350.1 Detailed Description

template<typename T, typename EXPR>
class Rcpp::sugar::Lazy< T, EXPR >

Definition at line 29 of file Lazy.h.

6.350.2 Member Function Documentation

6.350.2.1 template<typename T, typename EXPR> Rcpp::sugar::Lazy< T, EXPR >::operator T ( ) const [inline]

Definition at line 31 of file Lazy.h.

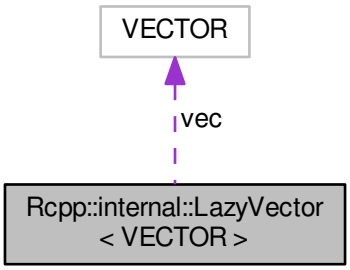
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/Lazy.h

6.351 Rcpp::internal::LazyVector< VECTOR > Class Template Reference

`#include <LazyVector.h>`

Collaboration diagram for Rcpp::internal::LazyVector< VECTOR >:



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

6.351.1 Detailed Description

template<typename VECTOR>
class Rcpp::internal::LazyVector< VECTOR >

Definition at line 29 of file LazyVector.h.

6.351.2 Member Typedef Documentation

6.351.2.1 template<typename VECTOR> typedef VECTOR::r_type Rcpp::internal::LazyVector< VECTOR >::r_type

Definition at line 31 of file LazyVector.h.

6.351.2.2 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.

6.351.3 Constructor & Destructor Documentation

6.351.3.1 template<typename VECTOR> Rcpp::internal::LazyVector< VECTOR >::LazyVector (const VECTOR & *vec_*) [inline]

Definition at line 34 of file LazyVector.h.

6.351.4 Member Function Documentation

6.351.4.1 template<typename VECTOR> stored_type Rcpp::internal::LazyVector< VECTOR >::operator[] (R_xlen_t *i*) const [inline]

Definition at line 36 of file LazyVector.h.

6.351.5 Member Data Documentation

6.351.5.1 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< RHS_TYPE >::operator[]().

6.351.5.2 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< RHS_TYPE >::operator[]().

6.351.5.3 template<typename VECTOR> R_xlen_t Rcpp::internal::LazyVector< VECTOR >::n [private]

Definition at line 49 of file LazyVector.h.

6.351.5.4 template<typename VECTOR> const VECTOR& Rcpp::internal::LazyVector< VECTOR >::vec [private]

Definition at line 48 of file LazyVector.h.

Referenced by Rcpp::internal::LazyVector< RHS_TYPE >::operator[](), and Rcpp::internal::LazyVector< Rcpp::↩ Vector< RTYPE > >::operator[]().

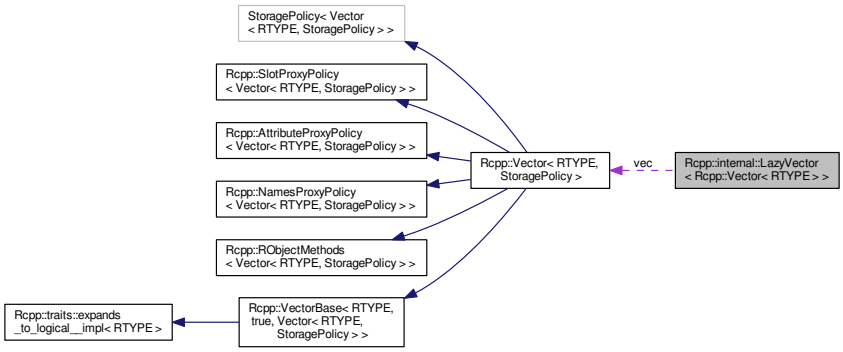
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

6.352 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

6.352.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >

Definition at line 55 of file LazyVector.h.

6.352.2 Member Typedef Documentation

6.352.2.1 template<int RTYPE> typedef VECTOR::Proxy Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::Proxy

Definition at line 58 of file LazyVector.h.

6.352.2.2 template<int RTYPE> typedef Rcpp::Vector<RTYPE> Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::VECTOR

Definition at line 57 of file LazyVector.h.

6.352.3 Constructor & Destructor Documentation

6.352.3.1 template<int RTYPE> Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::LazyVector (const VECTOR & *vec_*) [inline]

Definition at line 60 of file LazyVector.h.

6.352.4 Member Function Documentation

6.352.4.1 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.

6.352.5 Member Data Documentation

6.352.5.1 template<int RTYPE> const VECTOR& Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::vec [private]

Definition at line 64 of file LazyVector.h.

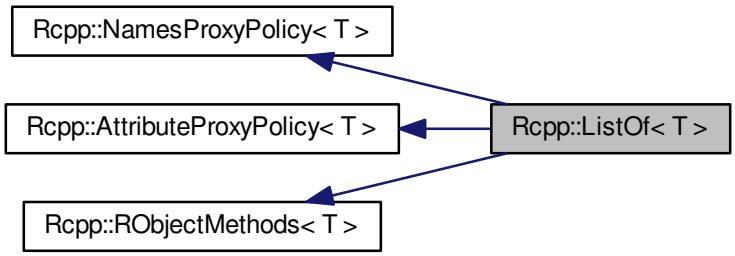
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

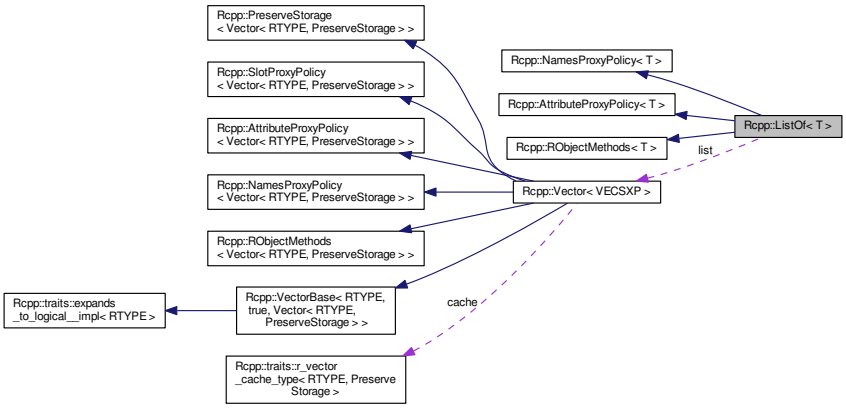
6.353 Rcpp::ListOf< T > Class Template Reference

`#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

6.353.1 Detailed Description

template<typename T>
class Rcpp::ListOf< T >

Definition at line 26 of file ListOf.h.

6.353.2 Member Typedef Documentation

6.353.2.1 template<typename T> typedef traits::r_vector_const_iterator<VECSXP>::type Rcpp::ListOf< T >::const_iterator

Definition at line 34 of file ListOf.h.

6.353.2.2 template<typename T> typedef traits::r_vector_iterator<VECSXP>::type Rcpp::ListOf< T >::iterator

Definition at line 33 of file ListOf.h.

6.353.3 Constructor & Destructor Documentation

6.353.3.1 template<typename T> Rcpp::ListOf< T >::ListOf ( ) [inline]

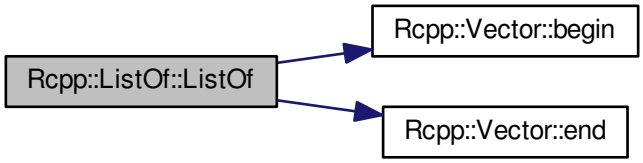
Definition at line 36 of file ListOf.h.

6.353.3.2 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:

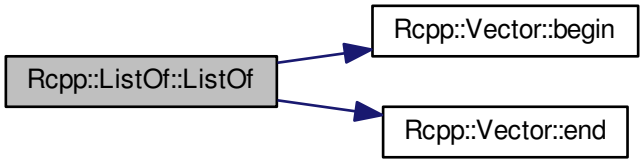


6.353.3.3 template<typename T> template<typename U > Rcpp::ListOf< T >::ListOf (const U & *data_*) [inline]

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:



6.353.3.4 template<typename T> Rcpp::ListOf< T >::ListOf (const ListOf< T > & *other*) [inline]

Definition at line 47 of file ListOf.h.

6.353.4 Member Function Documentation

6.353.4.1 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:

6.353.4.2 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:

6.353.4.3 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:

6.353.4.4 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:

6.353.4.5 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().

6.353.4.6 template<typename T> Rcpp::ListOf< T >::operator List ( ) const [inline]

Definition at line 108 of file ListOf.h.

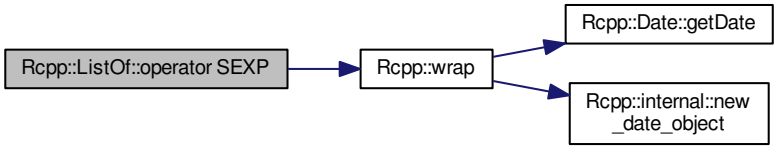
References Rcpp::ListOf< T >::list.

6.353.4.7 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:



6.353.4.8 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.

6.353.4.9 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.

6.353.4.10 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.

6.353.4.11 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.

6.353.4.12 template<typename T> ChildVector<T> Rcpp::ListOf< T >::operator[ ] (const std::string & *str*) [inline]

Definition at line 72 of file ListOf.h.

References Rcpp::ListOf< T >::list.

6.353.4.13 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::ListOf< T >::list.

6.353.4.14 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:

6.353.5 Member Data Documentation

6.353.5.1 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().

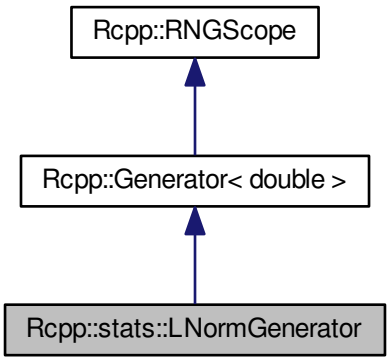
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ListOf.h

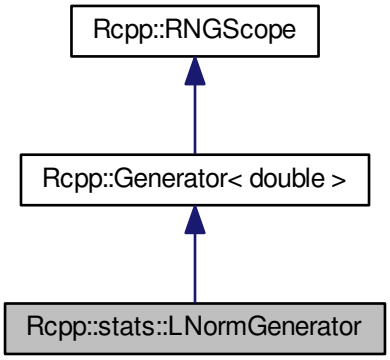
6.354 Rcpp::stats::LNormGenerator Class Reference

`#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

6.354.1 Detailed Description

Definition at line 28 of file rlnorm.h.

6.354.2 Constructor & Destructor Documentation

6.354.2.1 Rcpp::stats::LNormGenerator::LNormGenerator (double *meanlog_* = 0.0, double *sdlog_* = 1.0) [inline]

Definition at line 31 of file rlnorm.h.

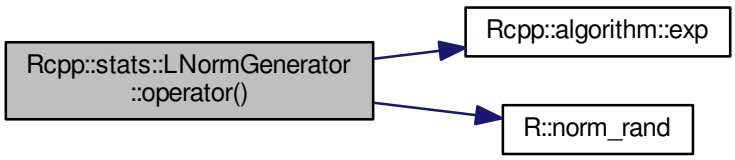
6.354.3 Member Function Documentation

6.354.3.1 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:



6.354.4 Member Data Documentation

6.354.4.1 double Rcpp::stats::LNormGenerator::meanlog [private]

Definition at line 39 of file rlnorm.h.

Referenced by operator()(), and Rcpp::stats::LNormGenerator_1::operator()().

6.354.4.2 double Rcpp::stats::LNormGenerator::sdlog [private]

Definition at line 40 of file rlnorm.h.

Referenced by operator()().

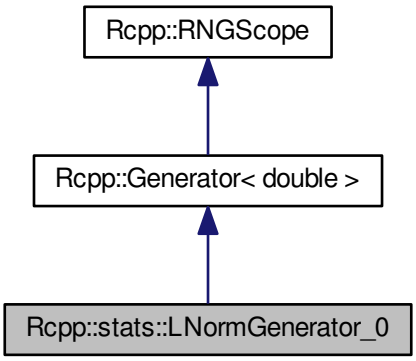
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

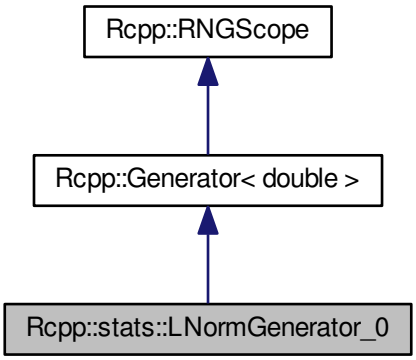
6.355 Rcpp::stats::LNormGenerator_0 Class Reference

```
#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

6.355.1 Detailed Description

Definition at line 59 of file rlnorm.h.

6.355.2 Constructor & Destructor Documentation

6.355.2.1 Rcpp::stats::LNormGenerator_0::LNormGenerator_0 ( ) [inline]

Definition at line 62 of file rlnorm.h.

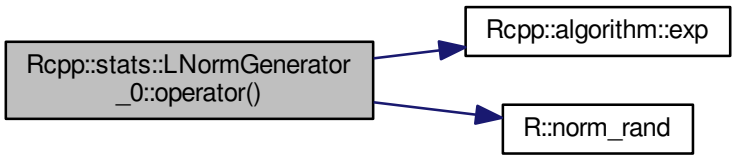
6.355.3 Member Function Documentation

6.355.3.1 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:



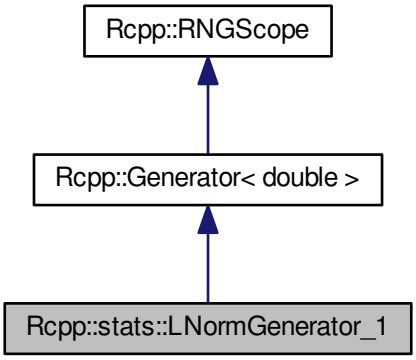
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

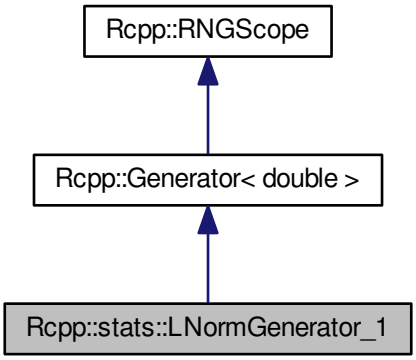
6.356 Rcpp::stats::LNormGenerator_1 Class Reference

`#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

6.356.1 Detailed Description

Definition at line 44 of file rlnorm.h.

6.356.2 Constructor & Destructor Documentation

6.356.2.1 Rcpp::stats::LNormGenerator_1::LNormGenerator_1 (double *meanlog_* = 0.0) [inline]

Definition at line 47 of file rlnorm.h.

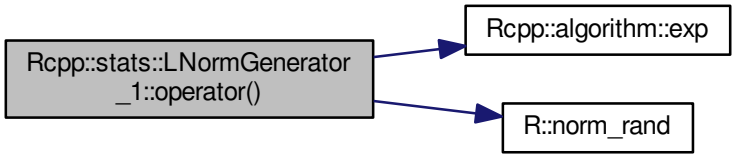
6.356.3 Member Function Documentation

6.356.3.1 double Rcpp::stats::LNormGenerator_1::operator() ( ) const [inline]

Definition at line 50 of file rlnorm.h.

References Rcpp::algorithm::exp(), Rcpp::stats::LNormGenerator::meanlog, and R::norm_rand().

Here is the call graph for this function:



6.356.4 Member Data Documentation

6.356.4.1 double Rcpp::stats::LNormGenerator_1::meanlog [private]

Definition at line 55 of file rlnorm.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

6.357 Rcpp::algorithm::helpers::log Struct Reference

`#include <algorithm.h>`

Public Member Functions

- template<typename T >
  double operator() (T val)

6.357.1 Detailed Description

Definition at line 176 of file algorithm.h.

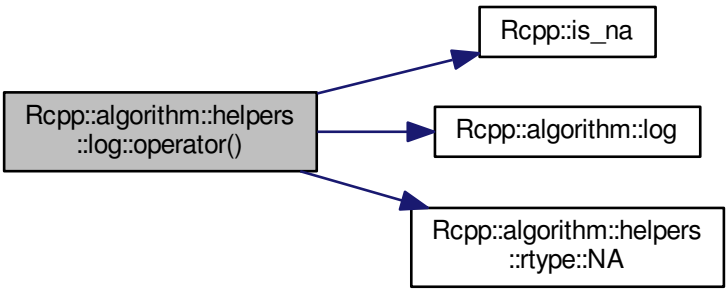
6.357.2 Member Function Documentation

6.357.2.1 template<typename T > double Rcpp::algorithm::helpers::log::operator() (T *val*) [inline]

Definition at line 178 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:



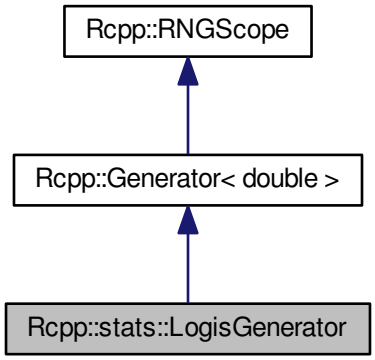
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

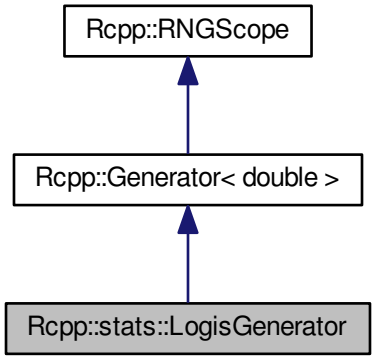
6.358 Rcpp::stats::LogisGenerator Class Reference

```
#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

6.358.1 Detailed Description

Definition at line 28 of file rlogis.h.

6.358.2 Constructor & Destructor Documentation

6.358.2.1 Rcpp::stats::LogisGenerator::LogisGenerator (double *location_*, double *scale_*) [inline]

Definition at line 31 of file rlogis.h.

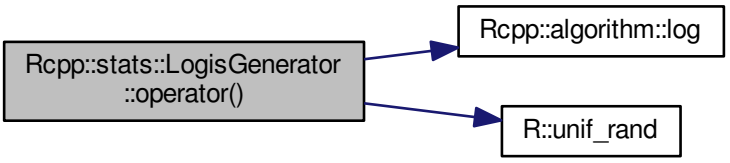
6.358.3 Member Function Documentation

6.358.3.1 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:



6.358.4 Member Data Documentation

6.358.4.1 double Rcpp::stats::LogisGenerator::location [private]

Definition at line 40 of file rlogis.h.

Referenced by operator()(), and Rcpp::stats::LogisGenerator_1::operator()().

6.358.4.2 double Rcpp::stats::LogisGenerator::scale [private]

Definition at line 41 of file rlogis.h.

Referenced by operator()().

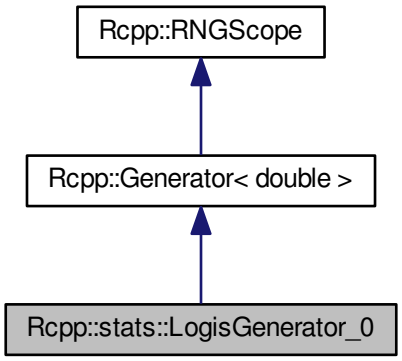
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

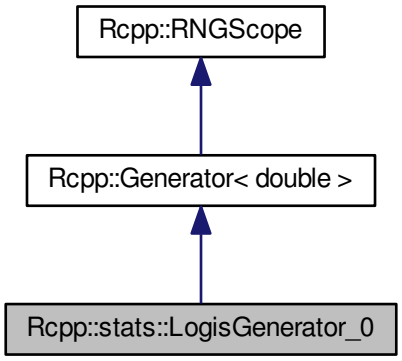
6.359 Rcpp::stats::LogisGenerator_0 Class Reference

```
#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

6.359.1 Detailed Description

Definition at line 61 of file rlogis.h.

6.359.2 Constructor & Destructor Documentation

6.359.2.1 Rcpp::stats::LogisGenerator_0::LogisGenerator_0 ( ) [inline]

Definition at line 64 of file rlogis.h.

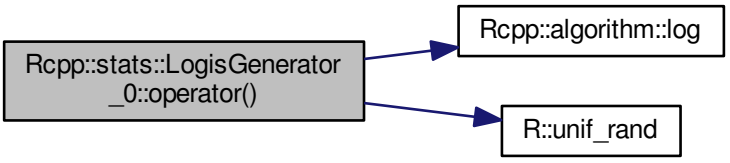
6.359.3 Member Function Documentation

6.359.3.1 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:



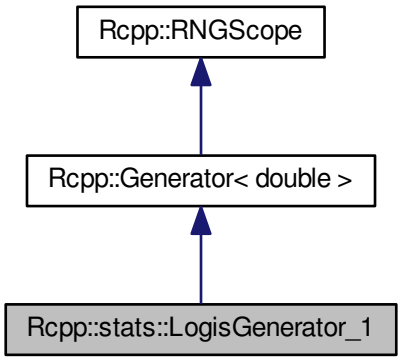
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

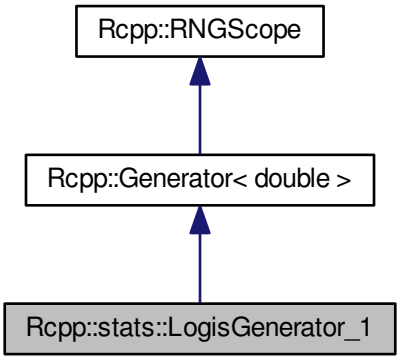
6.360 Rcpp::stats::LogisGenerator_1 Class Reference

`#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

6.360.1 Detailed Description

Definition at line 45 of file rlogis.h.

6.360.2 Constructor & Destructor Documentation

6.360.2.1 Rcpp::stats::LogisGenerator_1::LogisGenerator_1 (double *location_*) [inline]

Definition at line 48 of file rlogis.h.

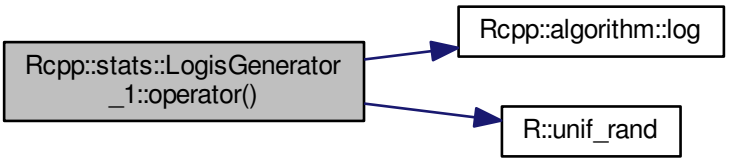
6.360.3 Member Function Documentation

6.360.3.1 double Rcpp::stats::LogisGenerator_1::operator() ( ) const [inline]

Definition at line 51 of file rlogis.h.

References Rcpp::stats::LogisGenerator::location, Rcpp::algorithm::log(), and R::unif_rand().

Here is the call graph for this function:



6.360.4 Member Data Documentation

6.360.4.1 double Rcpp::stats::LogisGenerator_1::location [private]

Definition at line 57 of file rlogis.h.

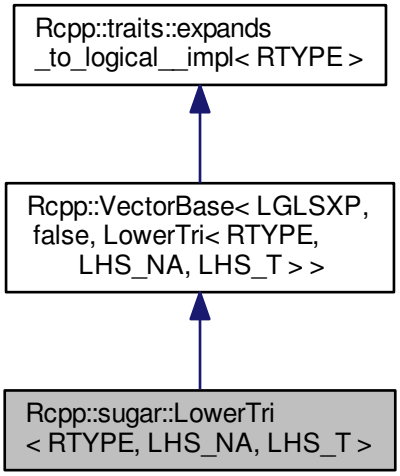
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

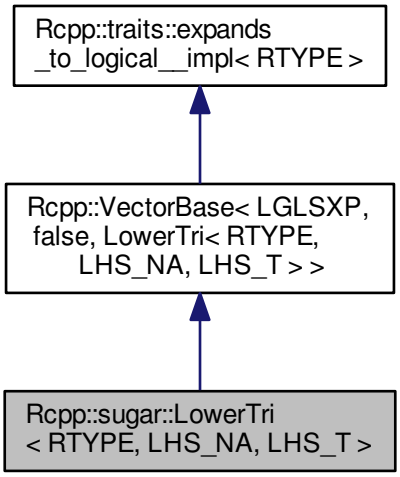
6.361 Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T > Class Template Reference

#include <lower_tri.h>

Inheritance diagram for Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >:



Collaboration diagram for Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >:



Public Types

- typedef Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE

Public Member Functions

- LowerTri (const LHS_TYPE &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)

Private Member Functions

- bool get_diag_true (int i, int j)
- bool get_diag_false (int i, int j)
- bool get (int i, int j)

Private Attributes

- int nr
- int nc
- Method getter

6.361.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >

Definition at line 29 of file lower_tri.h.

6.361.2 Member Typedef Documentation

6.361.2.1 template<int RTYPE, bool LHS_NA, typename LHS_T> typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::LHS_TYPE

Definition at line 35 of file lower_tri.h.

6.361.2.2 template<int RTYPE, bool LHS_NA, typename LHS_T> typedef bool(LowerTri::* Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::Method) (int, int) [private]

Definition at line 56 of file lower_tri.h.

6.361.3 Constructor & Destructor Documentation

6.361.3.1 template<int RTYPE, bool LHS_NA, typename LHS_T> Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::LowerTri (const LHS_TYPE & *lhs*, bool *diag*) [inline]

Definition at line 37 of file lower_tri.h.

6.361.4 Member Function Documentation

6.361.4.1 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::get (int *i*, int *j*) [inline], [private]

Definition at line 65 of file lower_tri.h.

6.361.4.2 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::get_diag_false (int *i*, int *j*) [inline], [private]

Definition at line 62 of file lower_tri.h.

6.361.4.3 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::get_diag_true (int *i*, int *j*) [inline], [private]

Definition at line 59 of file lower_tri.h.

6.361.4.4 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::ncol ( ) const [inline]

Definition at line 52 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nc.

6.361.4.5 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nrow ( ) const [inline]

Definition at line 51 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nr.

6.361.4.6 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::operator() (int *i*, int *j*) const [inline]

Definition at line 46 of file lower_tri.h.

6.361.4.7 template<int RTYPE, bool LHS_NA, typename LHS_T> R_xlen_t Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]

Definition at line 50 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nc, and Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nr.

6.361.5 Member Data Documentation

6.361.5.1 template<int RTYPE, bool LHS_NA, typename LHS_T> Method Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::getter [private]

Definition at line 58 of file lower_tri.h.

6.361.5.2 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nc [private]

Definition at line 55 of file lower_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::ncol(), and Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::size().

6.361.5.3 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nr [private]

Definition at line 55 of file lower_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::nrow(), and Rcpp::sugar::LowerTri< RTYPE, LHS_NA, LHS_T >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/lower_tri.h

6.362 Rcpp::lsinfo Struct Reference

Public Attributes

- time_t ls_trans
- long ls_corr

6.362.1 Detailed Description

Definition at line 329 of file Date.cpp.

6.362.2 Member Data Documentation

6.362.2.1 long Rcpp::lsinfo::ls_corr

Definition at line 331 of file Date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

6.362.2.2 time_t Rcpp::lsinfo::ls_trans

Definition at line 330 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

6.363 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container > Struct Template Reference

`#include <cbind.h>`

6.363.1 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

6.364 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true > Struct Template 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

6.364.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >

Definition at line 409 of file cbind.h.

6.364.2 Member Typedef Documentation

6.364.2.1 typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >::stored_type

Definition at line 411 of file cbind.h.

6.364.2.2 typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >::T

Definition at line 410 of file cbind.h.

6.364.3 Member Enumeration Documentation

6.364.3.1 anonymous enum

Enumerator

RTYPE

Definition at line 413 of file cbind.h.

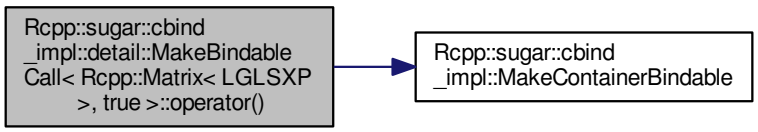
6.364.4 Member Function Documentation

6.364.4.1 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

6.365 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true > Struct Template 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

6.365.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >

Definition at line 421 of file cbind.h.

6.365.2 Member Typedef Documentation

6.365.2.1 typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >::stored_type

Definition at line 423 of file cbind.h.

6.365.2.2 typedef Rcpp::Vector<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >::T

Definition at line 422 of file cbind.h.

6.365.3 Member Enumeration Documentation

6.365.3.1 anonymous enum

Enumerator

RTYPE

Definition at line 425 of file cbind.h.

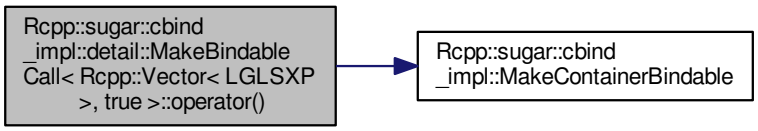
6.365.4 Member Function Documentation

6.365.4.1 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

6.366 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

6.366.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >

Definition at line 433 of file cbind.h.

6.366.2 Member Enumeration Documentation

6.366.2.1 template<typename T > anonymous enum

Enumerator

RTYPE

Definition at line 434 of file cbind.h.

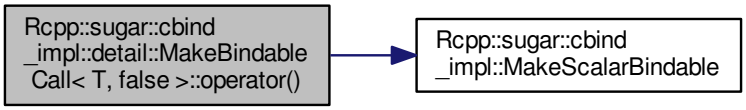
6.366.3 Member Function Documentation

6.366.3.1 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

6.367 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

6.367.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >

Definition at line 395 of file cbind.h.

6.367.2 Member Typedef Documentation

6.367.2.1 template<typename T > typedef cbind_storage_type< cbind_sexptype_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >::stored_type

Definition at line 398 of file cbind.h.

6.367.3 Member Enumeration Documentation

6.367.3.1 template<typename T > anonymous enum

Enumerator

RTYPE

Definition at line 400 of file cbind.h.

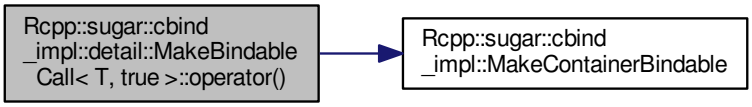
6.367.4 Member Function Documentation

6.367.4.1 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:



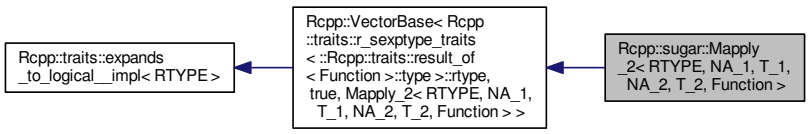
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

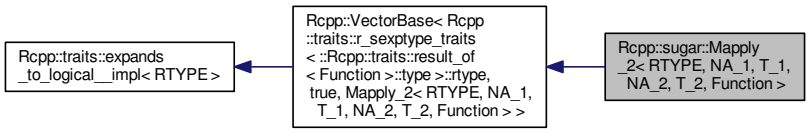
6.368 Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > Class Template Reference

`#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

6.368.1 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.

6.368.2 Member Typedef Documentation

6.368.2.1 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.

6.368.3 Constructor & Destructor Documentation

6.368.3.1 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.

6.368.4 Member Function Documentation

6.368.4.1 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.

6.368.4.2 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.

6.368.5 Member Data Documentation

6.368.5.1 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::mapply(), Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[](), Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[](), and Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[]().

6.368.5.2 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[](), Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[](), Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::size(), and Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::size().

6.368.5.3 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[](), 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().

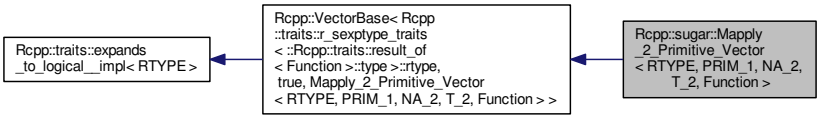
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

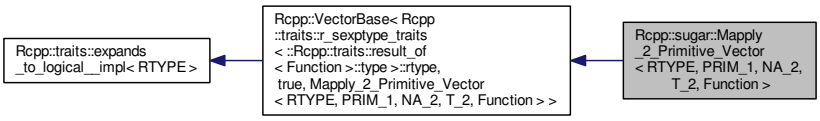
6.369 Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function > Class Template Reference

`#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

6.369.1 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.

6.369.2 Member Typedef Documentation

6.369.2.1 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.

6.369.3 Constructor & Destructor Documentation

6.369.3.1 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.

6.369.4 Member Function Documentation

6.369.4.1 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< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, and Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_2.

6.369.4.2 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< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_2.

6.369.5 Member Data Documentation

6.369.5.1 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.

6.369.5.2 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.

6.369.5.3 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.

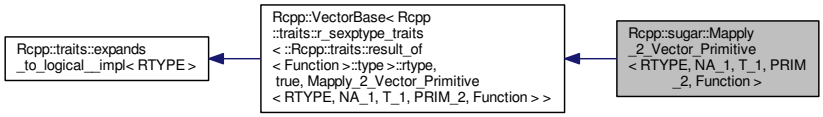
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

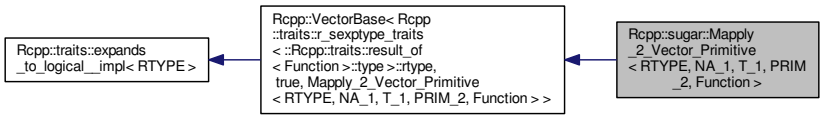
6.370 Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function > Class Template Reference

`#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

6.370.1 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.

6.370.2 Member Typedef Documentation

6.370.2.1 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.

6.370.3 Constructor & Destructor Documentation

6.370.3.1 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.

6.370.4 Member Function Documentation

6.370.4.1 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< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, and Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1.

6.370.4.2 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< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1.

6.370.5 Member Data Documentation

6.370.5.1 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.

6.370.5.2 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.

6.370.5.3 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.

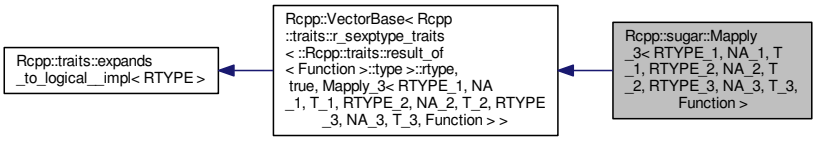
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

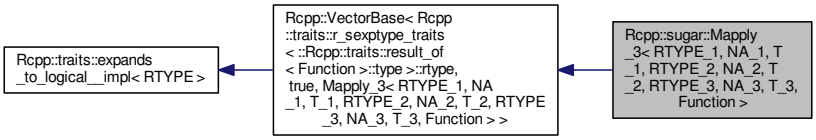
6.371 Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > Class Template Reference

`#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

6.371.1 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.

6.371.2 Member Typedef Documentation

6.371.2.1 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.

6.371.2.2 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.

6.371.2.3 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.

6.371.2.4 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> 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.

6.371.2.5 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.

6.371.2.6 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.

6.371.2.7 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.

6.371.3 Constructor & Destructor Documentation

6.371.3.1 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> 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.

6.371.4 Member Function Documentation

6.371.4.1 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> 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.

6.371.4.2 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> 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.

6.371.5 Member Data Documentation

6.371.5.1 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> 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::mapply(), and Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[ ]().

6.371.5.2 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_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().

6.371.5.3 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_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[]().

6.371.5.4 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> 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[]().

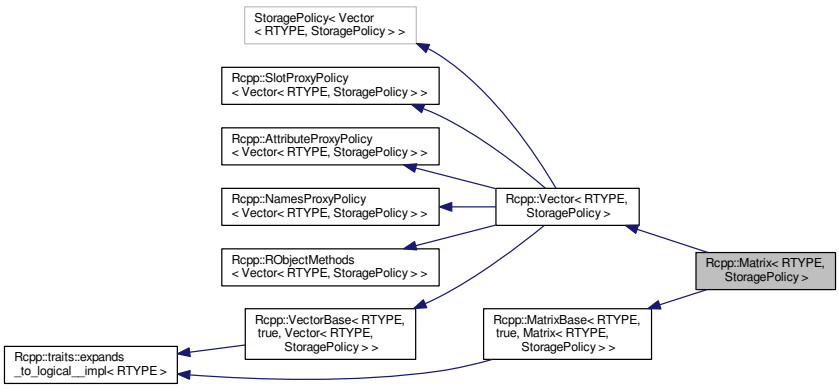
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_3.h

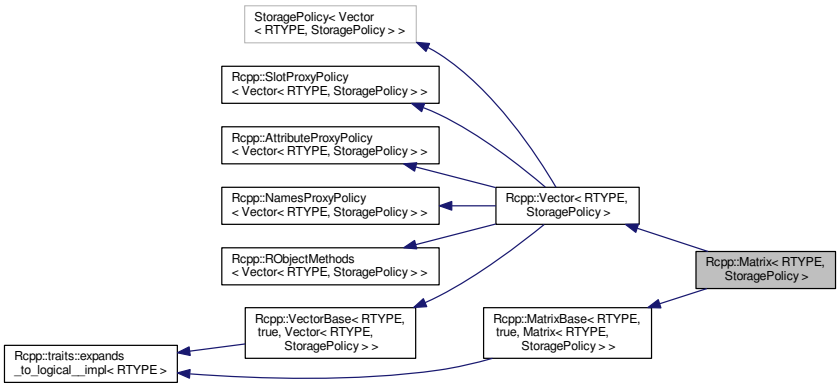
6.372 Rcpp::Matrix< RTYPE, StoragePolicy > Class Template Reference

`#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
- 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

6.372.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::Matrix< RTYPE, StoragePolicy >

Definition at line 28 of file Matrix.h.

6.372.2 Member Typedef Documentation

6.372.2.1 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef MatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Column

Definition at line 36 of file Matrix.h.

6.372.2.2 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::const_iterator

Definition at line 43 of file Matrix.h.

6.372.2.3 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::const_Proxy

Definition at line 47 of file Matrix.h.

6.372.2.4 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef ConstMatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstColumn

Definition at line 37 of file Matrix.h.

6.372.2.5 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef ConstMatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstRow

Definition at line 35 of file Matrix.h.

6.372.2.6 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::converter_type Rcpp::Matrix< RTYPE, StoragePolicy >::converter_type

Definition at line 44 of file Matrix.h.

6.372.2.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::iterator Rcpp::Matrix< RTYPE, StoragePolicy >::iterator

Definition at line 42 of file Matrix.h.

6.372.2.8 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::Proxy

Definition at line 46 of file Matrix.h.

6.372.2.9 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef MatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Row

Definition at line 34 of file Matrix.h.

6.372.2.10 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef StoragePolicy<Matrix> Rcpp::Matrix< RTYPE, StoragePolicy >::Storage

Definition at line 40 of file Matrix.h.

6.372.2.11 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef VECTOR::stored_type Rcpp::Matrix< RTYPE, StoragePolicy >::stored_type

Definition at line 45 of file Matrix.h.

6.372.2.12 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef SubMatrix<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Sub

Definition at line 38 of file Matrix.h.

6.372.2.13 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef Vector<RTYPE, StoragePolicy> Rcpp::Matrix< RTYPE, StoragePolicy >::VECTOR

Definition at line 41 of file Matrix.h.

6.372.3 Constructor & Destructor Documentation

6.372.3.1 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix ( ) [inline]

Definition at line 49 of file Matrix.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix().

6.372.3.2 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (SEXP *x*) [inline]

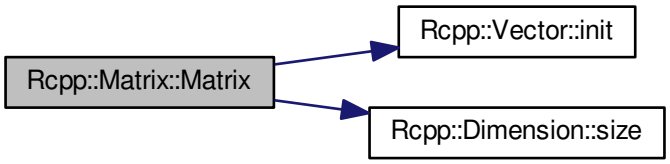
Definition at line 51 of file Matrix.h.

6.372.3.3 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const Dimension & *dims*) [inline]

Definition at line 53 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init(), and Rcpp::Dimension::size().

Here is the call graph for this function:



6.372.3.4 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const int & *nrows_*, const int & *ncols*) [inline]

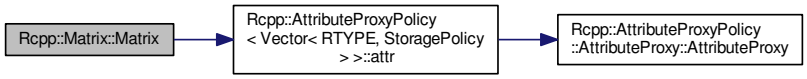
Definition at line 57 of file Matrix.h.

6.372.3.5 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 62 of file Matrix.h.

References Rcpp::AttributeProxyPolicy< Vector< RTYPE, StoragePolicy > >::attr().

Here is the call graph for this function:



6.372.3.6 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const int & *n*) [inline]

Definition at line 69 of file Matrix.h.

6.372.3.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const Matrix< RTYPE, StoragePolicy > & *other*) [inline]

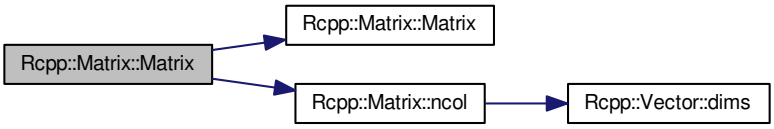
Definition at line 72 of file Matrix.h.

6.372.3.8 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 75 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:

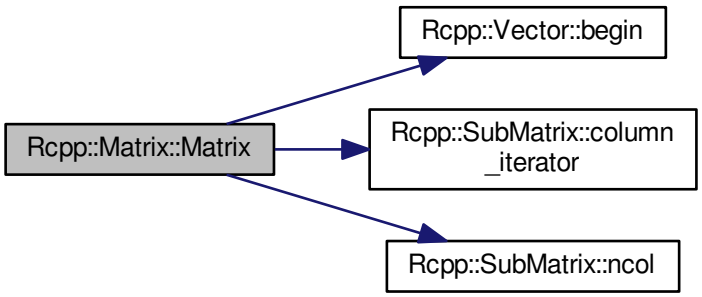


6.372.3.9 template<int RTYPE, template< class > class StoragePolicy> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const SubMatrix< RTYPE > & *sub*)

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:

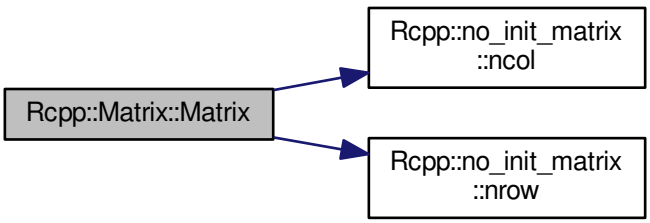


6.372.3.10 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (const no_init_matrix & *obj*) [inline], [explicit]

Definition at line 90 of file Matrix.h.

References Rcpp::no_init_matrix::ncol(), and Rcpp::no_init_matrix::nrow().

Here is the call graph for this function:



6.372.4 Member Function Documentation

6.372.4.1 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 142 of file Matrix.h.

6.372.4.2 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 145 of file Matrix.h.

6.372.4.3 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:

6.372.4.4 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Matrix< RTYPE, StoragePolicy >::begin ( ) [inline]

Definition at line 114 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

6.372.4.5 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:

6.372.4.6 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.

6.372.4.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::column (int *i*) const [inline]

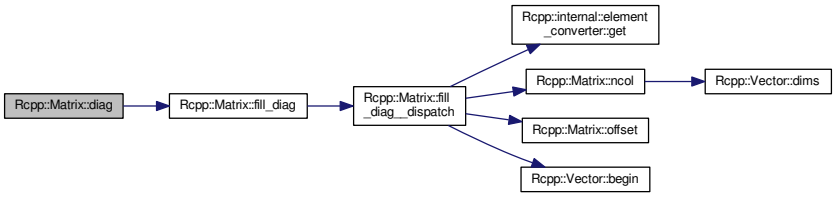
Definition at line 110 of file Matrix.h.

6.372.4.8 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 122 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag().

Here is the call graph for this function:



6.372.4.9 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:

6.372.4.10 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Matrix< RTYPE, StoragePolicy >::end ( ) [inline]

Definition at line 115 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:

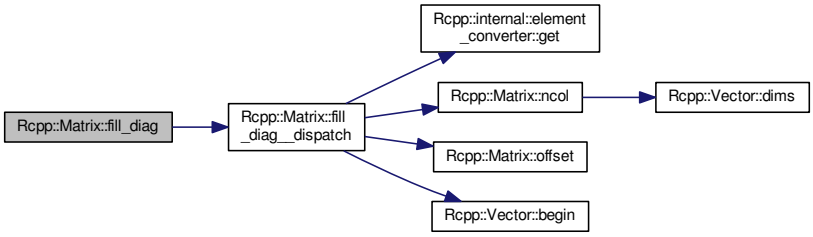
6.372.4.11 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename U > void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag (const U & *u*) [inline]

Definition at line 118 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:



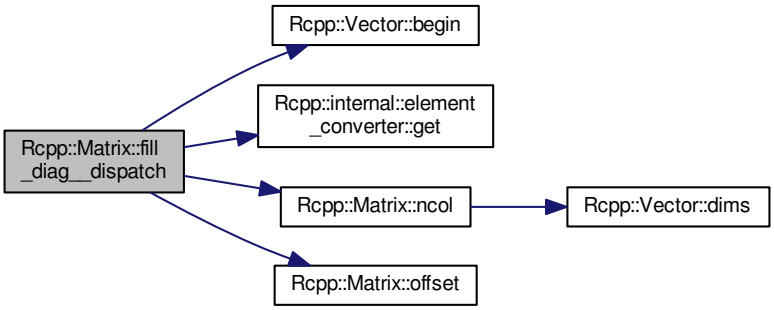
6.372.4.12 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename U > void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch (traits::false_type , const U & *u*) [inline], [private]

Definition at line 175 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::internal::element_converter< RTYPE >::get(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag().

Here is the call graph for this function:

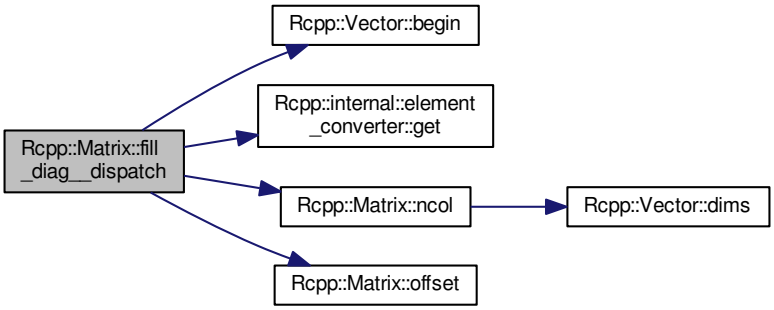


6.372.4.13 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename U > void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch (traits::true_type , const U & *u*) [inline], [private]

Definition at line 187 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::internal::element_converter< RTYPE >::get(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:

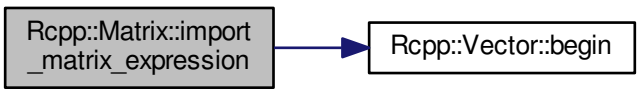


6.372.4.14 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<bool NA, typename MAT > void Rcpp::Matrix< RTYPE, StoragePolicy >::import_matrix_expression (const MatrixBase< RTYPE, NA, MAT > & *other,* int *nr,* int *nc*) [inline], [private]

Definition at line 199 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.372.4.15 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(), Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn(), fastLm(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::MatrixColumn< RTYPE >::MatrixColumn(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator()(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), Rcpp::MatrixRow< RTYPE >::size(), Rcpp::ConstMatrixRow< RTYPE >::size(), and Rcpp::tranpose_impl().

Here is the call graph for this function:

6.372.4.16 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::MatrixRow< RTYPE >::MatrixRow(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator()(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::tranpose_impl().

6.372.4.17 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 172 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

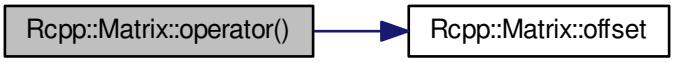
Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), and Rcpp::Matrix< RTYPE, StoragePolicy >::operator()().

6.372.4.18 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 135 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:

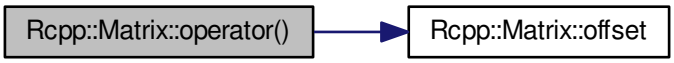


6.372.4.19 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (const size_t & *i*, const size_t & *j*) const [inline]

Definition at line 138 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



6.372.4.20 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Row Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (int *i*, internal::NamedPlaceHolder) [inline]

Definition at line 149 of file Matrix.h.

6.372.4.21 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> ConstRow Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (int *i*, internal::NamedPlaceHolder) const [inline]

Definition at line 152 of file Matrix.h.

6.372.4.22 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Column Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (internal::NamedPlaceHolder , int *i*) [inline]

Definition at line 155 of file Matrix.h.

6.372.4.23 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (internal::NamedPlaceHolder , int *i*) const [inline]

Definition at line 158 of file Matrix.h.

6.372.4.24 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (const Range & *row_range,* const Range & *col_range*) [inline]

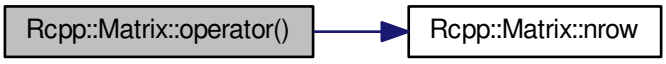
Definition at line 161 of file Matrix.h.

6.372.4.25 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (internal::NamedPlaceHolder , const Range & *col_range*) [inline]

Definition at line 164 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:

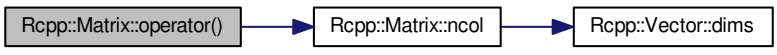


6.372.4.26 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (const Range & *row_range,* internal::NamedPlaceHolder) [inline]

Definition at line 167 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:



6.372.4.27 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Matrix& Rcpp::Matrix< RTYPE, StoragePolicy >::operator= (const Matrix< RTYPE, StoragePolicy > & *other*) [inline]

Definition at line 81 of file Matrix.h.

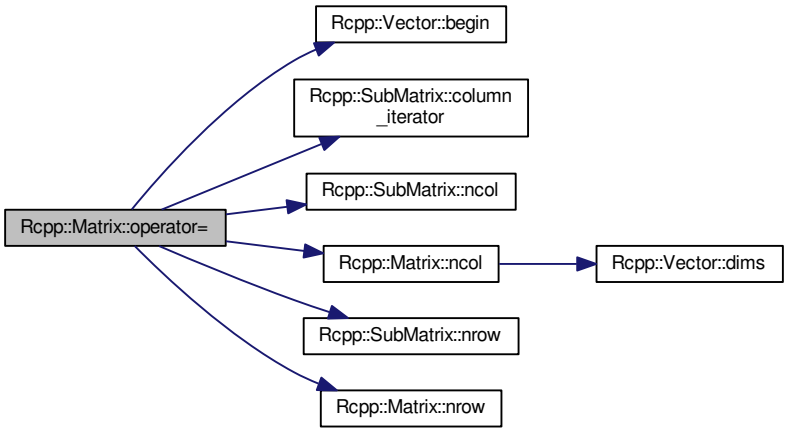
References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

6.372.4.28 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::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::SubMatrix< RTYPE >::column_iterator(), Rcpp::SubMatrix< RTYPE >::ncol(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), Rcpp::SubMatrix< RTYPE >::nrow(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:



6.372.4.29 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[ ] (R_xlen_t *i*) [inline]

Definition at line 128 of file Matrix.h.

6.372.4.30 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[ ] (R_xlen_t *i*) const [inline]

Definition at line 131 of file Matrix.h.

6.372.4.31 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.

Referenced by Rcpp::operator<<().

6.372.4.32 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.

6.372.4.33 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<<().

6.372.5 Member Data Documentation

6.372.5.1 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().

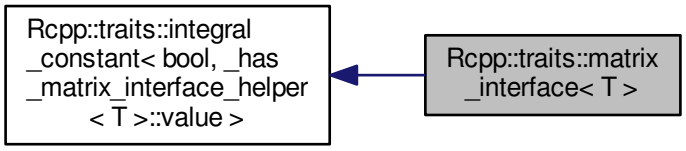
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/Matrix.h
- inst/include/Rcpp/vector/SubMatrix.h

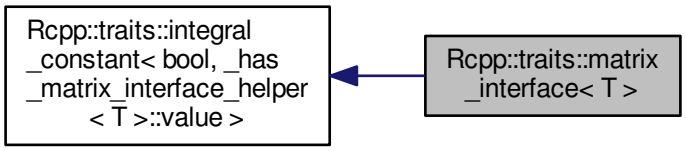
6.373 Rcpp::traits::matrix_interface< T > Struct Template Reference

`#include <matrix_interface.h>`

Inheritance diagram for Rcpp::traits::matrix_interface< T >:



Collaboration diagram for Rcpp::traits::matrix_interface< T >:



Additional Inherited Members

6.373.1 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

6.374 Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container > Struct Template Reference

`#include <cbind.h>`

6.374.1 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.

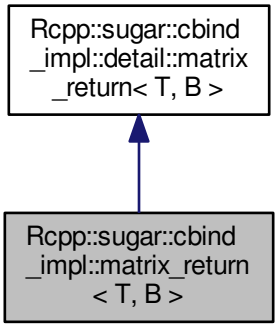
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

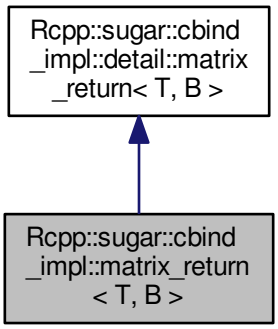
6.375 Rcpp::sugar::cbind_impl::matrix_return< T, B > Struct Template Reference

`#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 >:



6.375.1 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

6.376 Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false > Struct Template Reference

`#include <cbind.h>`

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.376.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false >

Definition at line 488 of file cbind.h.

6.376.2 Member Typedef Documentation

6.376.2.1 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

6.377 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true > Struct Template Reference

`#include <cbind.h>`

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.377.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true >

Definition at line 478 of file cbind.h.

6.377.2 Member Typedef Documentation

6.377.2.1 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

6.378 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true > Struct Template Reference

`#include <cbind.h>`

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.378.1 Detailed Description

template<>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true >

Definition at line 483 of file cbind.h.

6.378.2 Member Typedef Documentation

6.378.2.1 typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true >::type

Definition at line 484 of file cbind.h.

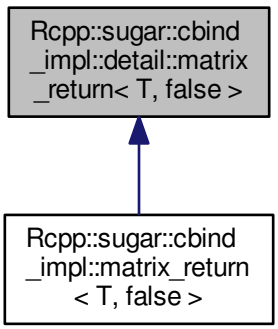
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.379 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

6.379.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >

Definition at line 493 of file cbind.h.

6.379.2 Member Typedef Documentation

6.379.2.1 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.

6.379.3 Member Enumeration Documentation

6.379.3.1 template<typename T > anonymous enum

Enumerator

RTYPE

Definition at line 494 of file cbind.h.

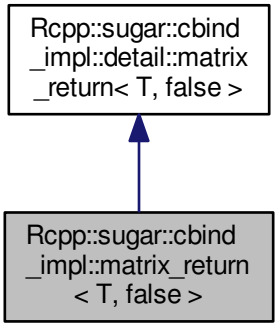
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

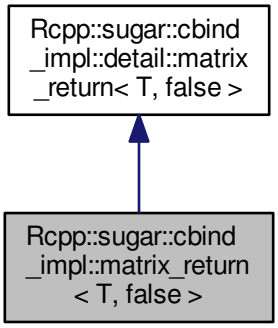
6.380 Rcpp::sugar::cbind_impl::matrix_return< T, false > Struct Template Reference

`#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

6.380.1 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

6.381 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

6.381.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >

Definition at line 466 of file cbind.h.

6.381.2 Member Typedef Documentation

6.381.2.1 template<typename T > typedef cbind_storage_type< cbind_sexptype_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >::stored_type

Definition at line 469 of file cbind.h.

6.381.2.2 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.

6.381.3 Member Enumeration Documentation

6.381.3.1 template<typename T > anonymous enum

Enumerator

RTYPE

Definition at line 471 of file cbind.h.

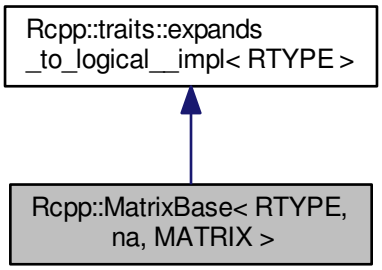
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

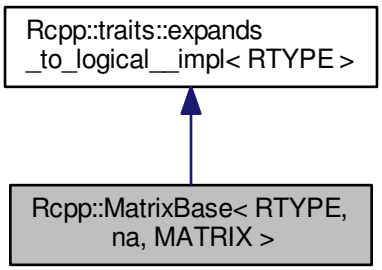
6.382 Rcpp::MatrixBase< RTYPE, na, MATRIX > Class Template Reference

`#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

6.382.1 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 31 of file MatrixBase.h.

6.382.2 Member Typedef Documentation

6.382.2.1 template<int RTYPE, bool na, typename MATRIX> typedef traits::storage_type<RTYPE>::type Rcpp::MatrixBase< RTYPE, na, MATRIX >::stored_type

Definition at line 36 of file MatrixBase.h.

6.382.3 Member Function Documentation

6.382.3.1 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::begin ( ) const [inline]

Definition at line 158 of file MatrixBase.h.

6.382.3.2 template<int RTYPE, bool na, typename MATRIX> iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::end ( ) const [inline]

Definition at line 159 of file MatrixBase.h.

6.382.3.3 template<int RTYPE, bool na, typename MATRIX> MATRIX& Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref ( ) [inline]

Definition at line 38 of file MatrixBase.h.

Referenced by Rcpp::internal::as_vector__impl().

6.382.3.4 template<int RTYPE, bool na, typename MATRIX> R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol ( ) const [inline]

Definition at line 48 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, false, T, false >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

6.382.3.5 template<int RTYPE, bool na, typename MATRIX> R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow ( ) const [inline]

Definition at line 47 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, false, T, false >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, false, T, false >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

6.382.3.6 template<int RTYPE, bool na, typename MATRIX> stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::operator() (int *i*, int *j*) const [inline]

Definition at line 42 of file MatrixBase.h.

6.382.3.7 template<int RTYPE, bool na, typename MATRIX> R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::size ( ) const [inline]

Definition at line 46 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, NA, T >::end().

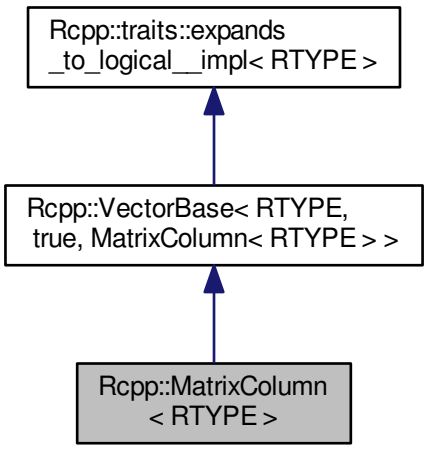
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

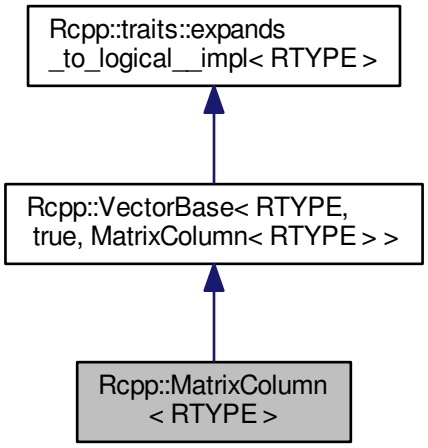
6.383 Rcpp::MatrixColumn< RTYPE > Class Template Reference

`#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
- iterator begin ()
- const_iterator begin () const
- iterator end ()
- const_iterator end () const
- int size () const

Private Attributes

- const int n
- iterator start
- const_iterator const_start

6.383.1 Detailed Description

template<int RTYPE>
class Rcpp::MatrixColumn< RTYPE >

Definition at line 28 of file MatrixColumn.h.

6.383.2 Member Typedef Documentation

6.383.2.1 template<int RTYPE> typedef MATRIX::const_iterator Rcpp::MatrixColumn< RTYPE >::const_iterator

Definition at line 35 of file MatrixColumn.h.

6.383.2.2 template<int RTYPE> typedef MATRIX::const_Proxy Rcpp::MatrixColumn< RTYPE >::const_Proxy

Definition at line 32 of file MatrixColumn.h.

6.383.2.3 template<int RTYPE> typedef MATRIX::iterator Rcpp::MatrixColumn< RTYPE >::iterator

Definition at line 34 of file MatrixColumn.h.

6.383.2.4 template<int RTYPE> typedef Matrix<RTYPE> Rcpp::MatrixColumn< RTYPE >::MATRIX

Definition at line 30 of file MatrixColumn.h.

6.383.2.5 template<int RTYPE> typedef MATRIX::Proxy Rcpp::MatrixColumn< RTYPE >::Proxy

Definition at line 31 of file MatrixColumn.h.

6.383.2.6 template<int RTYPE> typedef MATRIX::value_type Rcpp::MatrixColumn< RTYPE >::value_type

Definition at line 33 of file MatrixColumn.h.

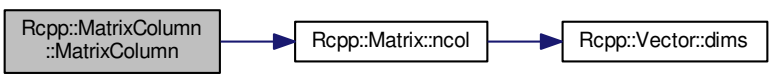
6.383.3 Constructor & Destructor Documentation

6.383.3.1 template<int RTYPE> Rcpp::MatrixColumn< RTYPE >::MatrixColumn (MATRIX & *parent,* int *i*) [inline]

Definition at line 37 of file MatrixColumn.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:

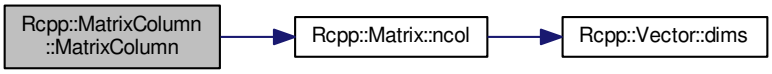


6.383.3.2 template<int RTYPE> Rcpp::MatrixColumn< RTYPE >::MatrixColumn (const MATRIX & *parent,* int *i*) [inline]

Definition at line 45 of file MatrixColumn.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:



6.383.3.3 template<int RTYPE> Rcpp::MatrixColumn< RTYPE >::MatrixColumn (const MatrixColumn< RTYPE > & *other*) [inline]

Definition at line 53 of file MatrixColumn.h.

6.383.4 Member Function Documentation

6.383.4.1 template<int RTYPE> iterator Rcpp::MatrixColumn< RTYPE >::begin ( ) [inline]

Definition at line 79 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

6.383.4.2 template<int RTYPE> const_iterator Rcpp::MatrixColumn< RTYPE >::begin ( ) const [inline]

Definition at line 83 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.383.4.3 template<int RTYPE> iterator Rcpp::MatrixColumn< RTYPE >::end ( ) [inline]

Definition at line 87 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n, and Rcpp::MatrixColumn< RTYPE >::start.

6.383.4.4 template<int RTYPE> const_iterator Rcpp::MatrixColumn< RTYPE >::end ( ) const [inline]

Definition at line 91 of file MatrixColumn.h.

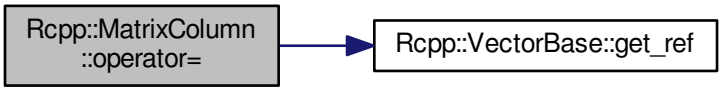
References Rcpp::MatrixColumn< RTYPE >::const_start, and Rcpp::MatrixColumn< RTYPE >::n.

6.383.4.5 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 59 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:



6.383.4.6 template<int RTYPE> MatrixColumn& Rcpp::MatrixColumn< RTYPE >::operator= (const MatrixColumn< RTYPE > & *rhs*) [inline]

Definition at line 65 of file MatrixColumn.h.

References RCPP_LOOP_UNROLL, and Rcpp::MatrixColumn< RTYPE >::start.

6.383.4.7 template<int RTYPE> Proxy Rcpp::MatrixColumn< RTYPE >::operator[ ] (int *i*) [inline]

Definition at line 71 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

6.383.4.8 template<int RTYPE> const_Proxy Rcpp::MatrixColumn< RTYPE >::operator[ ] (int *i*) const [inline]

Definition at line 75 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.383.4.9 template<int RTYPE> int Rcpp::MatrixColumn< RTYPE >::size ( ) const [inline]

Definition at line 95 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n.

6.383.5 Member Data Documentation

6.383.5.1 template<int RTYPE> const_iterator Rcpp::MatrixColumn< RTYPE >::const_start [private]

Definition at line 102 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::ConstMatrixColumn< RTYPE >::begin(), Rcpp::↩
MatrixColumn< RTYPE >::end(), Rcpp::ConstMatrixColumn< RTYPE >::end(), Rcpp::MatrixColumn< RTYPE >↩
::operator[ ](), and Rcpp::ConstMatrixColumn< RTYPE >::operator[ ]().

6.383.5.2 template<int RTYPE> const int Rcpp::MatrixColumn< RTYPE >::n [private]

Definition at line 100 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::end(), Rcpp::ConstMatrixColumn< RTYPE >::end(), Rcpp::Matrix↩
Column< RTYPE >::size(), and Rcpp::ConstMatrixColumn< RTYPE >::size().

6.383.5.3 template<int RTYPE> iterator Rcpp::MatrixColumn< RTYPE >::start [private]

Definition at line 101 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::MatrixColumn< RTYPE >::end(), Rcpp::Matrix↩
Column< RTYPE >::operator=(), and Rcpp::MatrixColumn< RTYPE >::operator[ ]().

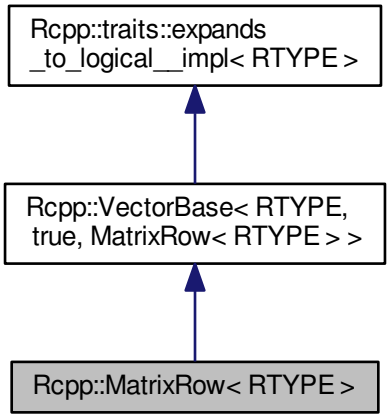
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

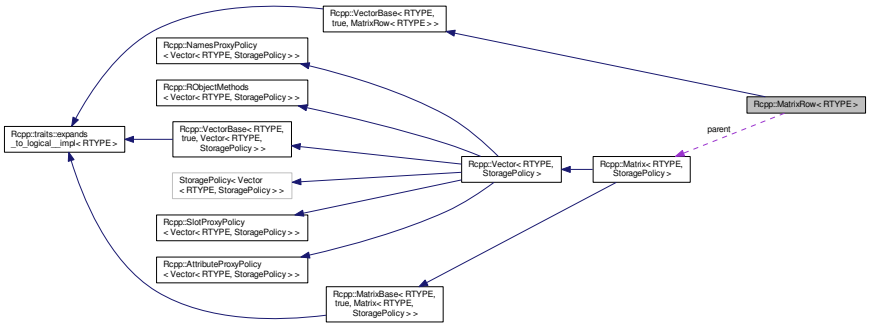
6.384 Rcpp::MatrixRow< RTYPE > Class Template Reference

`#include <MatrixRow.h>`

Inheritance diagram for Rcpp::MatrixRow< RTYPE >:



Collaboration diagram for Rcpp::MatrixRow< RTYPE >:



Classes

- class iterator

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

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 ()
- iterator begin () const
- iterator end () 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

6.384.1 Detailed Description

template<int RTYPE>
class Rcpp::MatrixRow< RTYPE >

Definition at line 28 of file MatrixRow.h.

6.384.2 Member Typedef Documentation

6.384.2.1 template<int RTYPE> typedef MATRIX::const_Proxy Rcpp::MatrixRow< RTYPE >::const_reference

Definition at line 33 of file MatrixRow.h.

6.384.2.2 template<int RTYPE> typedef Matrix<RTYPE> Rcpp::MatrixRow< RTYPE >::MATRIX

Definition at line 30 of file MatrixRow.h.

6.384.2.3 template<int RTYPE> typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::Proxy

Definition at line 31 of file MatrixRow.h.

6.384.2.4 template<int RTYPE> typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::reference

Definition at line 32 of file MatrixRow.h.

6.384.2.5 template<int RTYPE> typedef MATRIX::value_type Rcpp::MatrixRow< RTYPE >::value_type

Definition at line 34 of file MatrixRow.h.

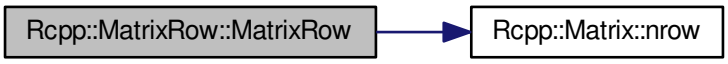
6.384.3 Constructor & Destructor Documentation

6.384.3.1 template<int RTYPE> Rcpp::MatrixRow< RTYPE >::MatrixRow (MATRIX & *object*, int *i*) [inline]

Definition at line 104 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::MatrixRow< RTYPE >::parent.

Here is the call graph for this function:



6.384.3.2 template<int RTYPE> Rcpp::MatrixRow< RTYPE >::MatrixRow (const MatrixRow< RTYPE > & *other*) [inline]

Definition at line 113 of file MatrixRow.h.

6.384.4 Member Function Documentation

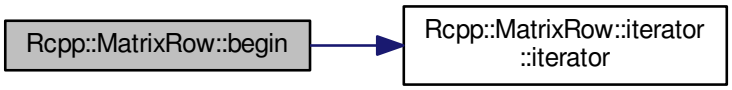
6.384.4.1 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::begin ( ) [inline]

Definition at line 142 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::iterator().

Referenced by Rcpp::operator<<().

Here is the call graph for this function:

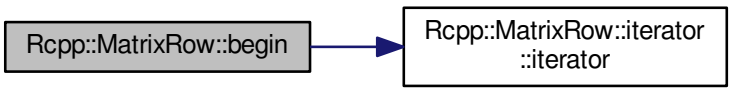


6.384.4.2 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::begin ( ) const [inline]

Definition at line 150 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::iterator().

Here is the call graph for this function:

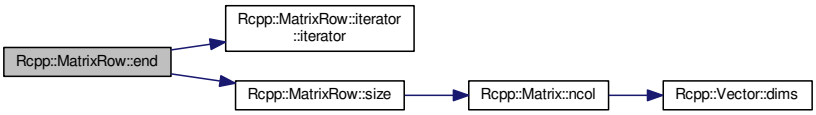


6.384.4.3 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::end ( ) [inline]

Definition at line 146 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:

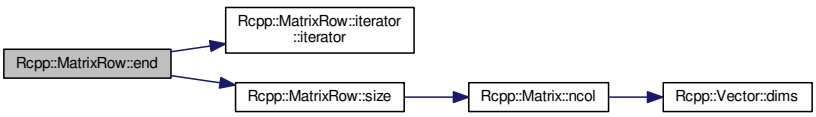


6.384.4.4 template<int RTYPE> iterator Rcpp::MatrixRow< RTYPE >::end ( ) const [inline]

Definition at line 154 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iterator::iterator(), and Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



6.384.4.5 template<int RTYPE> int Rcpp::MatrixRow< RTYPE >::get_parent_index (int *i*) const [inline], [private]

Definition at line 168 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent_nrow, and RCPP_DEBUG_4.

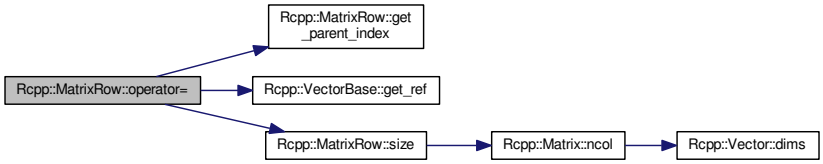
Referenced by Rcpp::MatrixRow< RTYPE >::operator=(), and Rcpp::MatrixRow< RTYPE >::operator[]().

6.384.4.6 template<int RTYPE> template<int RT, bool NA, typename T > MatrixRow& Rcpp::MatrixRow< RTYPE >::operator= (const Rcpp::VectorBase< RT, NA, T > & *rhs*) [inline]

Definition at line 121 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:

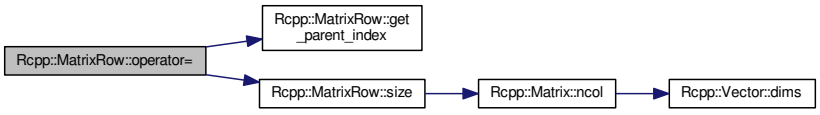


6.384.4.7 template<int RTYPE> MatrixRow& Rcpp::MatrixRow< RTYPE >::operator= (const MatrixRow< RTYPE > & *rhs*) [inline]

Definition at line 128 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:

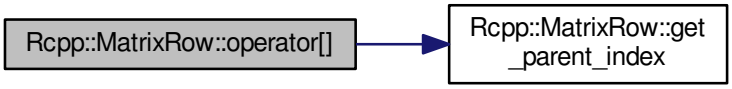


6.384.4.8 template<int RTYPE> reference Rcpp::MatrixRow< RTYPE >::operator[] (int *i*) [inline]

Definition at line 134 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get_parent_index(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



6.384.4.9 template<int RTYPE> reference Rcpp::MatrixRow< RTYPE >::operator[] (int *i*) const [inline]

Definition at line 138 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent, Rcpp::MatrixRow< RTYPE >::parent_nrow, and Rcpp::MatrixRow< RTYPE >::iterator::row.

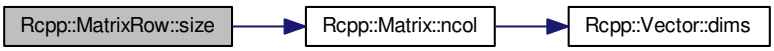
6.384.4.10 template<int RTYPE> int Rcpp::MatrixRow< RTYPE >::size ( ) const [inline]

Definition at line 158 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::MatrixRow< RTYPE >::parent.

Referenced by Rcpp::MatrixRow< RTYPE >::end(), Rcpp::ConstMatrixRow< RTYPE >::end(), and Rcpp::MatrixRow< RTYPE >::operator=().

Here is the call graph for this function:



6.384.5 Member Data Documentation

6.384.5.1 template<int RTYPE> MATRIX& Rcpp::MatrixRow< RTYPE >::parent [private]

Definition at line 163 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow(), Rcpp::MatrixRow< RTYPE >::MatrixRow(), Rcpp::MatrixRow< RTYPE >::operator[](), Rcpp::ConstMatrixRow< RTYPE >::operator[](), Rcpp::MatrixRow< RTYPE >::size(), and Rcpp::ConstMatrixRow< RTYPE >::size().

6.384.5.2 template<int RTYPE> int Rcpp::MatrixRow< RTYPE >::parent_nrow [private]

Definition at line 165 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::get_parent_index(), Rcpp::ConstMatrixRow< RTYPE >::get_parent_↩ index(), Rcpp::MatrixRow< RTYPE >::operator[](), and Rcpp::ConstMatrixRow< RTYPE >::operator[]().

6.384.5.3 template<int RTYPE> int Rcpp::MatrixRow< RTYPE >::row [private]

Definition at line 166 of file MatrixRow.h.

6.384.5.4 template<int RTYPE> MATRIX::iterator Rcpp::MatrixRow< RTYPE >::start [private]

Definition at line 164 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

6.385 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

6.385.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Max< RTYPE, NA, T >

Definition at line 29 of file max.h.

6.385.2 Member Typedef Documentation

6.385.2.1 template<int RTYPE, bool NA, typename T > typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Max< RTYPE, NA, T >::STORAGE

Definition at line 31 of file max.h.

6.385.3 Constructor & Destructor Documentation

6.385.3.1 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Max< RTYPE, NA, T >::Max (const T & *obj_*) [inline]

Definition at line 33 of file max.h.

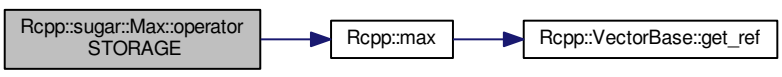
6.385.4 Member Function Documentation

6.385.4.1 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE ( ) const [inline]

Definition at line 35 of file max.h.

References Rcpp::max(), and Rcpp::sugar::Max< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.385.5 Member Data Documentation

6.385.5.1 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

6.386 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

6.386.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Max< RTYPE, false, T >

Definition at line 55 of file max.h.

6.386.2 Member Typedef Documentation

6.386.2.1 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.

6.386.3 Constructor & Destructor Documentation

6.386.3.1 template<int RTYPE, typename T > Rcpp::sugar::Max< RTYPE, false, T >::Max (const T & *obj_*) [inline]

Definition at line 59 of file max.h.

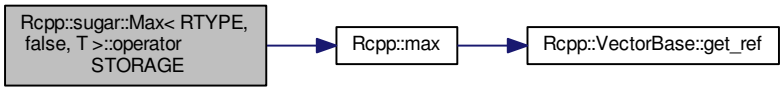
6.386.4 Member Function Documentation

6.386.4.1 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:



6.386.5 Member Data Documentation

6.386.5.1 template<int RTYPE, typename T > const T& Rcpp::sugar::Max< RTYPE, false, T >::obj [private]

Definition at line 74 of file max.h.

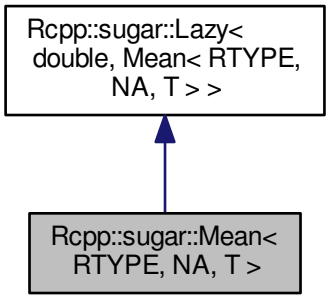
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/max.h

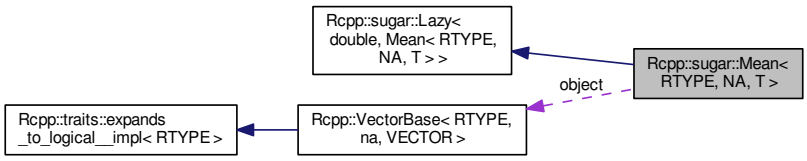
6.387 Rcpp::sugar::Mean< RTYPE, NA, T > Class Template Reference

`#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

6.387.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Mean< RTYPE, NA, T >

Definition at line 29 of file mean.h.

6.387.2 Member Typedef Documentation

6.387.2.1 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.

6.387.2.2 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.

6.387.3 Constructor & Destructor Documentation

6.387.3.1 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.

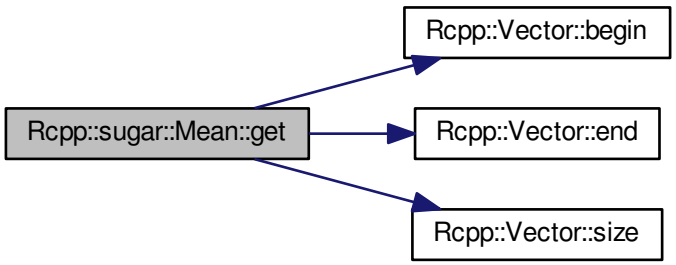
6.387.4 Member Function Documentation

6.387.4.1 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:



6.387.5 Member Data Documentation

6.387.5.1 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().

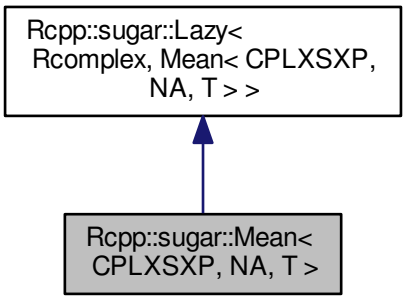
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

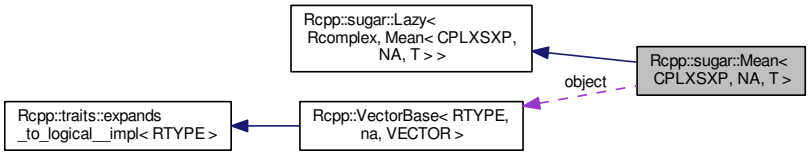
6.388 Rcpp::sugar::Mean< CPLXSXP, NA, T > Class Template Reference

`#include <mean.h>`

Inheritance diagram for Rcpp::sugar::Mean< CPLXSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Mean< CPLXSXP, NA, T >:



Public Types

- typedef Rcpp::VectorBase< CPLXSXP, NA, T > VEC_TYPE

Public Member Functions

- Mean (const VEC_TYPE &object_)
- Rcomplex get () const

Private Attributes

- const VEC_TYPE & object

6.388.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< CPLXSXP, NA, T >

Definition at line 55 of file mean.h.

6.388.2 Member Typedef Documentation

6.388.2.1 template<bool NA, typename T > typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::Mean< CPLXSXP, NA, T >::VEC_TYPE

Definition at line 57 of file mean.h.

6.388.3 Constructor & Destructor Documentation

6.388.3.1 template<bool NA, typename T > Rcpp::sugar::Mean< CPLXSXP, NA, T >::Mean (const VEC_TYPE & *object_*) [inline]

Definition at line 59 of file mean.h.

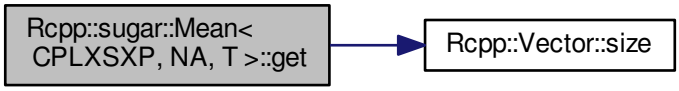
6.388.4 Member Function Documentation

6.388.4.1 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:



6.388.5 Member Data Documentation

6.388.5.1 template<bool NA, typename T > const VEC_TYPE& Rcpp::sugar::Mean< CPLXSXP, NA, T >::object [private]

Definition at line 88 of file mean.h.

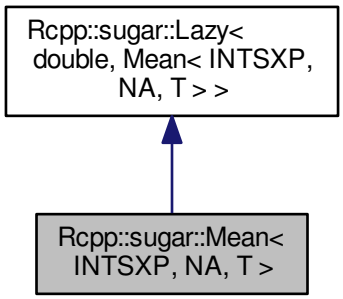
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

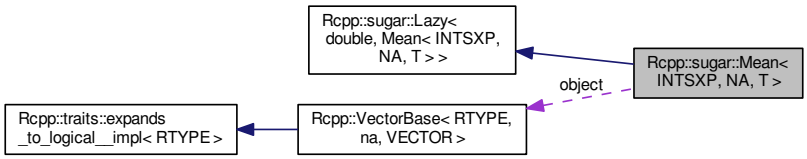
6.389 Rcpp::sugar::Mean< INTSXP, NA, T > Class Template Reference

`#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

6.389.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< INTSXP, NA, T >

Definition at line 114 of file mean.h.

6.389.2 Member Typedef Documentation

6.389.2.1 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.

6.389.3 Constructor & Destructor Documentation

6.389.3.1 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.

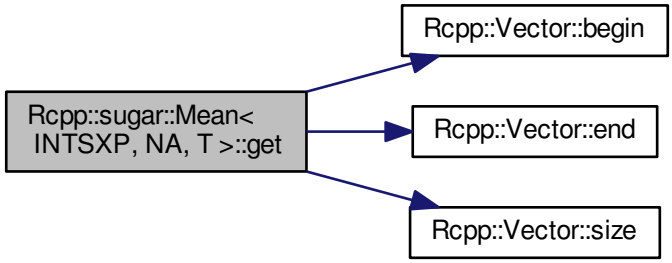
6.389.4 Member Function Documentation

6.389.4.1 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:



6.389.5 Member Data Documentation

6.389.5.1 template<bool NA, typename T > const VEC_TYPE& Rcpp::sugar::Mean< INTSXP, NA, T >::object [private]

Definition at line 134 of file mean.h.

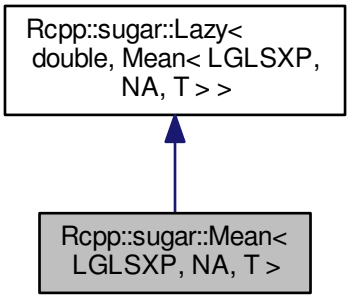
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

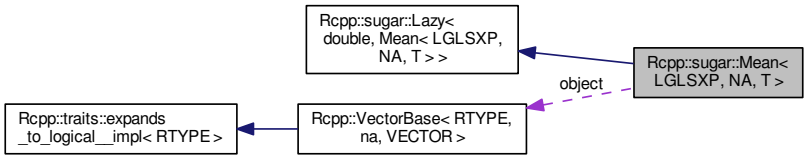
6.390 Rcpp::sugar::Mean< LGLSXP, NA, T > Class Template Reference

```
#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

6.390.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< LGLSXP, NA, T >

Definition at line 92 of file mean.h.

6.390.2 Member Typedef Documentation

6.390.2.1 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.

6.390.3 Constructor & Destructor Documentation

6.390.3.1 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.

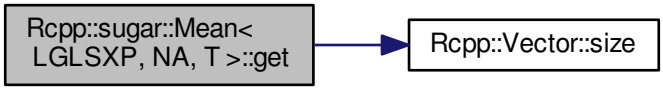
6.390.4 Member Function Documentation

6.390.4.1 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:



6.390.5 Member Data Documentation

6.390.5.1 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

6.391 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

6.391.1 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.

6.391.2 Member Typedef Documentation

6.391.2.1 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.

6.391.2.2 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.

6.391.2.3 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.

6.391.3 Member Enumeration Documentation

6.391.3.1 template<int RTYPE, bool NA, typename T , bool NA_RM = false> anonymous enum

Enumerator

RESULT_RTYPE

Definition at line 91 of file median.h.

6.391.4 Constructor & Destructor Documentation

6.391.4.1 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.

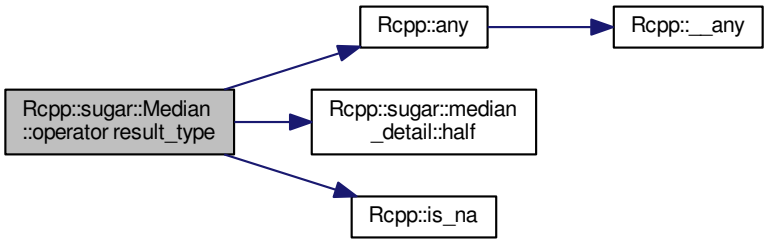
6.391.5 Member Function Documentation

6.391.5.1 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(), and Rcpp::is_na().

Here is the call graph for this function:



6.391.6 Member Data Documentation

6.391.6.1 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.392 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

6.392.1 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.

6.392.2 Member Typedef Documentation

6.392.2.1 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.

6.392.2.2 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.

6.392.2.3 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.

6.392.3 Member Enumeration Documentation

6.392.3.1 template<int RTYPE, typename T , bool NA_RM> anonymous enum

Enumerator

RESULT_RTYPE

Definition at line 163 of file median.h.

6.392.4 Constructor & Destructor Documentation

6.392.4.1 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.

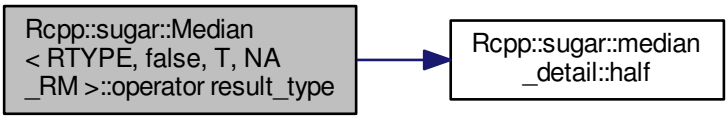
6.392.5 Member Function Documentation

6.392.5.1 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().

Here is the call graph for this function:



6.392.6 Member Data Documentation

6.392.6.1 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

6.393 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

6.393.1 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.

6.393.2 Member Typedef Documentation

6.393.2.1 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.

6.393.2.2 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.

6.393.2.3 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.

6.393.3 Member Enumeration Documentation

6.393.3.1 template<int RTYPE, bool NA, typename T > anonymous enum

Enumerator

RESULT_RTYPE

Definition at line 129 of file median.h.

6.393.4 Constructor & Destructor Documentation

6.393.4.1 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.

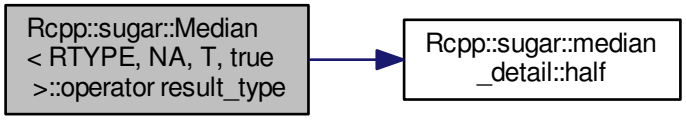
6.393.5 Member Function Documentation

6.393.5.1 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().

Here is the call graph for this function:



6.393.6 Member Data Documentation

6.393.6.1 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

6.394 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

6.394.1 Detailed Description

template<typename T>
class Rcpp::sugar::Median< STRSXP, false, T, true >

Definition at line 254 of file median.h.

6.394.2 Member Typedef Documentation

6.394.2.1 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.

6.394.2.2 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.

6.394.2.3 template<typename T > typedef T Rcpp::sugar::Median< STRSXP, false, T, true >::VECTOR

Definition at line 258 of file median.h.

6.394.3 Constructor & Destructor Documentation

6.394.3.1 template<typename T > Rcpp::sugar::Median< STRSXP, false, T, true >::Median (const VECTOR & *xx*) [inline]

Definition at line 264 of file median.h.

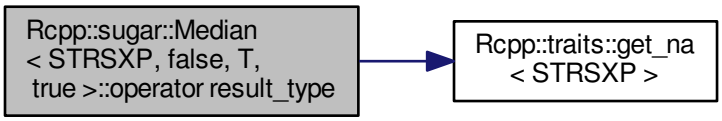
6.394.4 Member Function Documentation

6.394.4.1 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 >().

Here is the call graph for this function:



6.394.5 Member Data Documentation

6.394.5.1 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

6.395 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

6.395.1 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.

6.395.2 Member Typedef Documentation

6.395.2.1 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.

6.395.2.2 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.

6.395.2.3 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.

6.395.3 Constructor & Destructor Documentation

6.395.3.1 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.

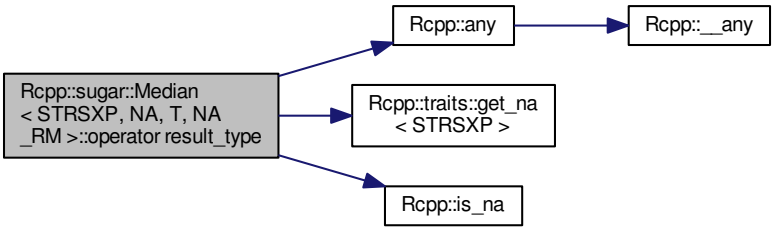
6.395.4 Member Function Documentation

6.395.4.1 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 >(), and Rcpp::is_na().

Here is the call graph for this function:



6.395.5 Member Data Documentation

6.395.5.1 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

6.396 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

6.396.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Median< STRSXP, NA, T, true >

Definition at line 227 of file median.h.

6.396.2 Member Typedef Documentation

6.396.2.1 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.

6.396.2.2 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.

6.396.2.3 template<bool NA, typename T > typedef T Rcpp::sugar::Median< STRSXP, NA, T, true >::VECTOR

Definition at line 231 of file median.h.

6.396.3 Constructor & Destructor Documentation

6.396.3.1 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.

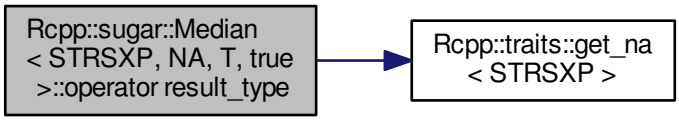
6.396.4 Member Function Documentation

6.396.4.1 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 >().

Here is the call graph for this function:



6.396.5 Member Data Documentation

6.396.5.1 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

6.397 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

6.397.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Min< RTYPE, NA, T >

Definition at line 29 of file min.h.

6.397.2 Member Typedef Documentation

6.397.2.1 template<int RTYPE, bool NA, typename T > typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, NA, T >::STORAGE

Definition at line 31 of file min.h.

6.397.3 Constructor & Destructor Documentation

6.397.3.1 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Min< RTYPE, NA, T >::Min (const T & *obj_*) [inline]

Definition at line 33 of file min.h.

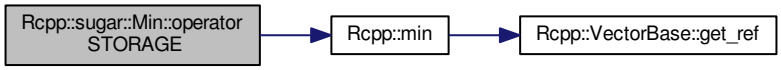
6.397.4 Member Function Documentation

6.397.4.1 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE ( ) const [inline]

Definition at line 35 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.397.5 Member Data Documentation

6.397.5.1 template<int RTYPE, bool NA, typename T > const T& Rcpp::sugar::Min< RTYPE, NA, T >::obj

Definition at line 49 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

6.398 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

6.398.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Min< RTYPE, false, T >

Definition at line 54 of file min.h.

6.398.2 Member Typedef Documentation

6.398.2.1 template<int RTYPE, typename T > typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, false, T >::STORAGE

Definition at line 56 of file min.h.

6.398.3 Constructor & Destructor Documentation

6.398.3.1 template<int RTYPE, typename T > Rcpp::sugar::Min< RTYPE, false, T >::Min (const T & *obj_*) [inline]

Definition at line 58 of file min.h.

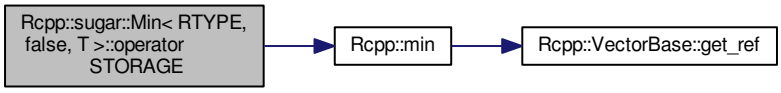
6.398.4 Member Function Documentation

6.398.4.1 template<int RTYPE, typename T > Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE ( ) const [inline]

Definition at line 60 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.398.5 Member Data Documentation

6.398.5.1 template<int RTYPE, typename T > const T& Rcpp::sugar::Min< RTYPE, false, T >::obj

Definition at line 72 of file min.h.

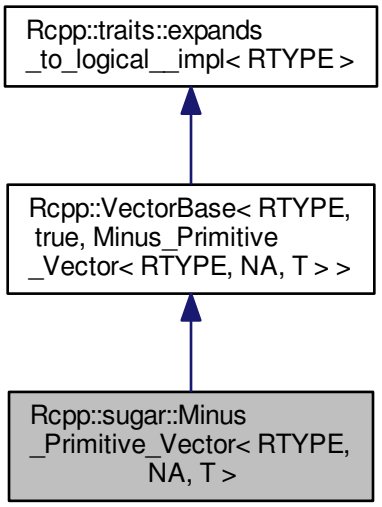
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/min.h

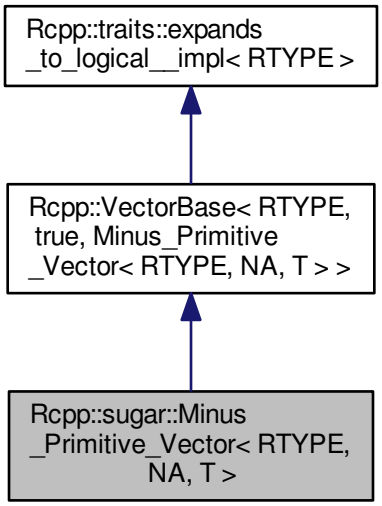
6.399 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T > Class Template Reference

`#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

6.399.1 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.

6.399.2 Member Typedef Documentation

6.399.2.1 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.

6.399.2.2 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.

6.399.2.3 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.

6.399.3 Constructor & Destructor Documentation

6.399.3.1 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.

6.399.4 Member Function Documentation

6.399.4.1 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_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.

6.399.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.399.5 Member Data Documentation

6.399.5.1 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.

6.399.5.2 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.

6.399.5.3 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.

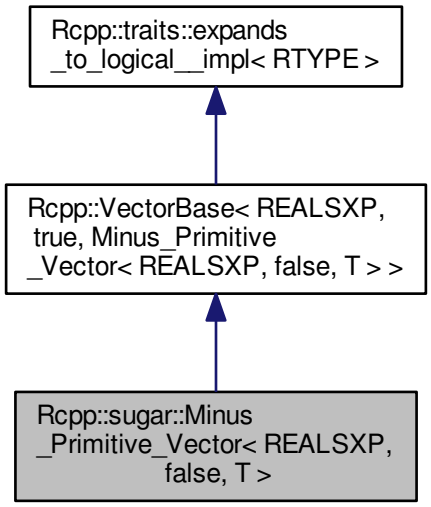
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

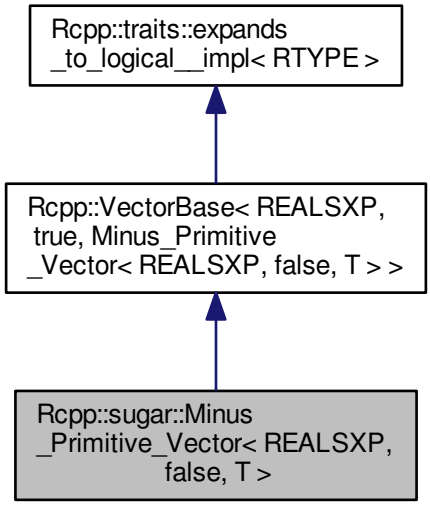
6.400 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T > Class Template Reference

`#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

6.400.1 Detailed Description

template<typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >

Definition at line 392 of file minus.h.

6.400.2 Member Typedef Documentation

6.400.2.1 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.

6.400.2.2 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.

6.400.3 Constructor & Destructor Documentation

6.400.3.1 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.

6.400.4 Member Function Documentation

6.400.4.1 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_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.

6.400.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.400.5 Member Data Documentation

6.400.5.1 template<typename T > double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::lhs [private]

Definition at line 408 of file minus.h.

6.400.5.2 template<typename T > const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::rhs [private]

Definition at line 409 of file minus.h.

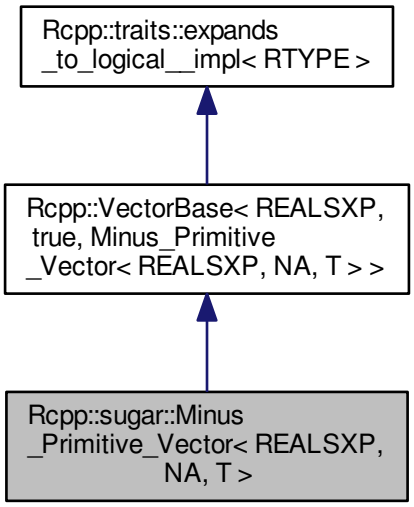
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

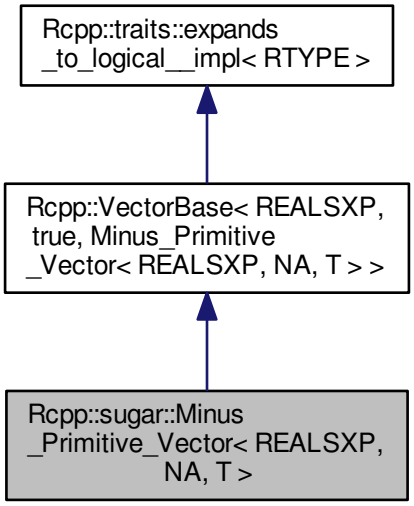
6.401 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T > Class Template Reference

`#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

6.401.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >

Definition at line 346 of file minus.h.

6.401.2 Member Typedef Documentation

6.401.2.1 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.

6.401.2.2 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.

6.401.3 Constructor & Destructor Documentation

6.401.3.1 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.

6.401.4 Member Function Documentation

6.401.4.1 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_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.

6.401.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.401.5 Member Data Documentation

6.401.5.1 template<bool NA, typename T > double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::lhs [private]

Definition at line 361 of file minus.h.

6.401.5.2 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.

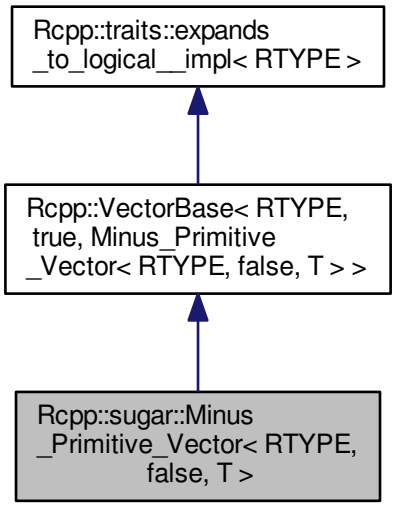
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

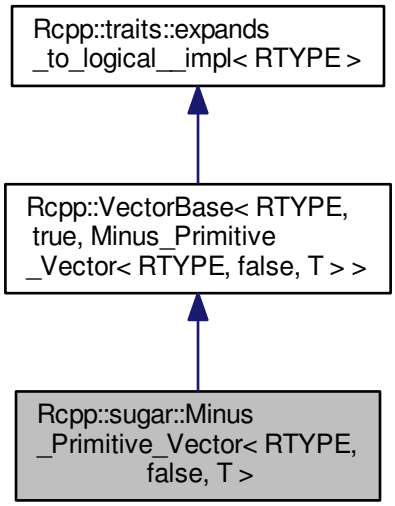
6.402 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T > Class Template Reference

`#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

6.402.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >

Definition at line 368 of file minus.h.

6.402.2 Member Typedef Documentation

6.402.2.1 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.

6.402.2.2 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.

6.402.2.3 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.

6.402.3 Constructor & Destructor Documentation

6.402.3.1 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.

6.402.4 Member Function Documentation

6.402.4.1 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_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.

6.402.4.2 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_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.402.5 Member Data Documentation

6.402.5.1 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::lhs [private]

Definition at line 386 of file minus.h.

6.402.5.2 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.

6.402.5.3 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.

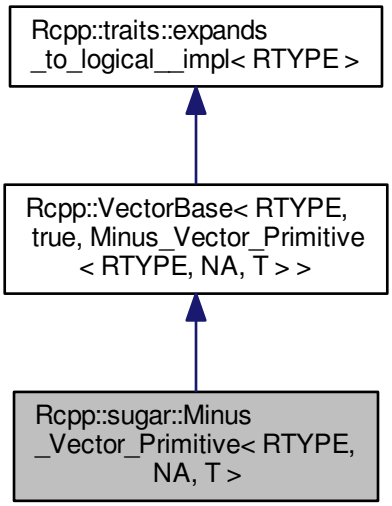
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

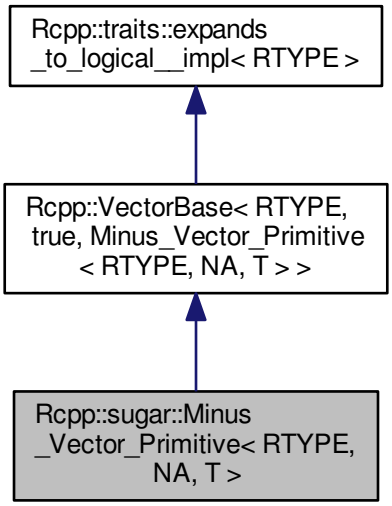
6.403 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T > Class Template Reference

`#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

6.403.1 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.

6.403.2 Member Typedef Documentation

6.403.2.1 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.

6.403.2.2 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.

6.403.2.3 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.

6.403.3 Constructor & Destructor Documentation

6.403.3.1 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.

6.403.4 Member Function Documentation

6.403.4.1 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_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.

6.403.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.403.5 Member Data Documentation

6.403.5.1 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.

6.403.5.2 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.

6.403.5.3 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.

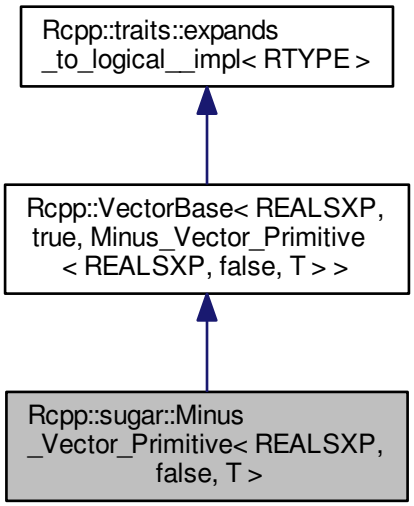
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

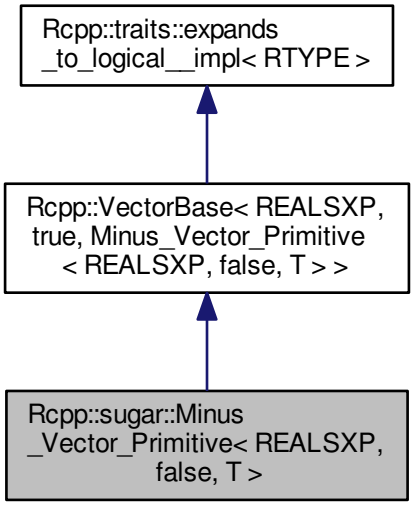
6.404 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T > Class Template Reference

`#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

6.404.1 Detailed Description

template<typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >

Definition at line 298 of file minus.h.

6.404.2 Member Typedef Documentation

6.404.2.1 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.

6.404.2.2 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.

6.404.3 Constructor & Destructor Documentation

6.404.3.1 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.

6.404.4 Member Function Documentation

6.404.4.1 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_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.

6.404.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.404.5 Member Data Documentation

6.404.5.1 template<typename T > const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::lhs [private]

Definition at line 314 of file minus.h.

6.404.5.2 template<typename T > double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::rhs [private]

Definition at line 315 of file minus.h.

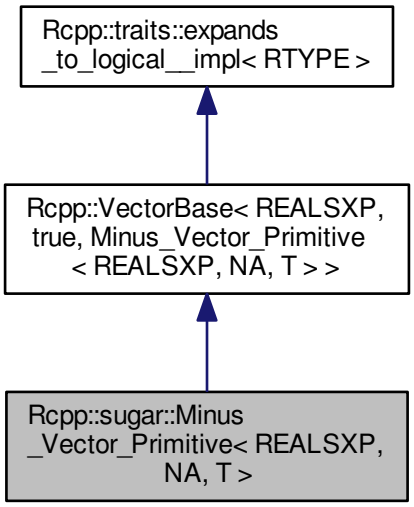
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

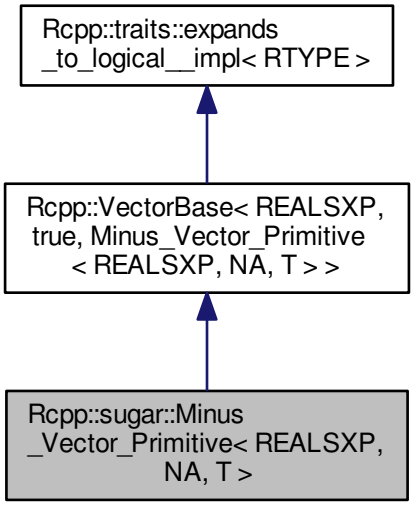
6.405 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T > Class Template Reference

```
#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

6.405.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >

Definition at line 252 of file minus.h.

6.405.2 Member Typedef Documentation

6.405.2.1 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.

6.405.2.2 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.

6.405.3 Constructor & Destructor Documentation

6.405.3.1 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.

6.405.4 Member Function Documentation

6.405.4.1 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_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.

6.405.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.405.5 Member Data Documentation

6.405.5.1 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.

6.405.5.2 template<bool NA, typename T > double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::rhs [private]

Definition at line 269 of file minus.h.

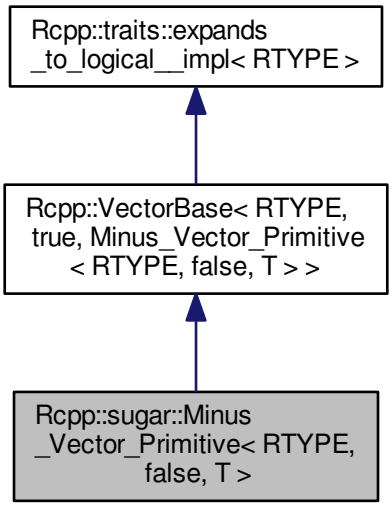
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

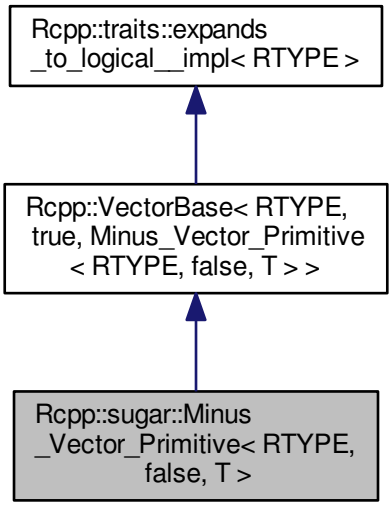
6.406 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T > Class Template Reference

`#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

6.406.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >

Definition at line 274 of file minus.h.

6.406.2 Member Typedef Documentation

6.406.2.1 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.

6.406.2.2 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.

6.406.2.3 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.

6.406.3 Constructor & Destructor Documentation

6.406.3.1 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.

6.406.4 Member Function Documentation

6.406.4.1 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_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.

6.406.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.406.5 Member Data Documentation

6.406.5.1 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.

6.406.5.2 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::rhs [private]

Definition at line 294 of file minus.h.

6.406.5.3 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.

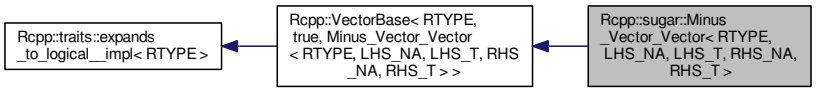
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

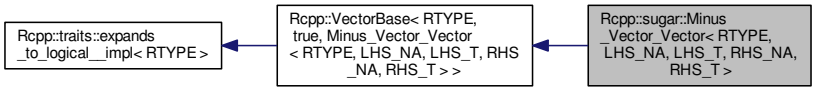
6.407 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.407.1 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.

6.407.2 Member Typedef Documentation

6.407.2.1 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.

6.407.2.2 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.

6.407.2.3 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.

6.407.2.4 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.

6.407.2.5 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.

6.407.3 Constructor & Destructor Documentation

6.407.3.1 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.

6.407.4 Member Function Documentation

6.407.4.1 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.

6.407.4.2 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.

6.407.5 Member Data Documentation

6.407.5.1 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::operator-(), 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_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_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_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(), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size(), 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().

6.407.5.2 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::operator-(), 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_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_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().

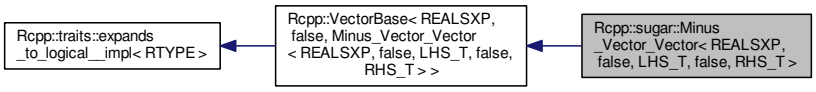
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

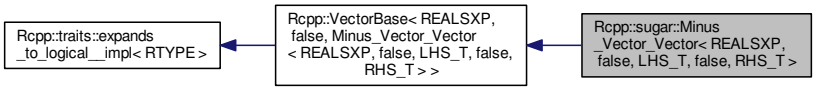
6.408 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.408.1 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.

6.408.2 Member Typedef Documentation

6.408.2.1 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.

6.408.2.2 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.

6.408.2.3 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.

6.408.2.4 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.

6.408.3 Constructor & Destructor Documentation

6.408.3.1 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.

6.408.4 Member Function Documentation

6.408.4.1 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.

6.408.4.2 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.

6.408.5 Member Data Documentation

6.408.5.1 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.

6.408.5.2 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.

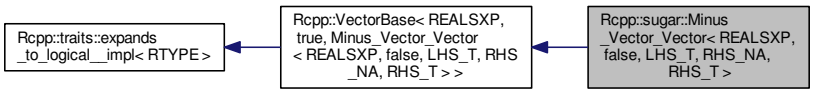
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

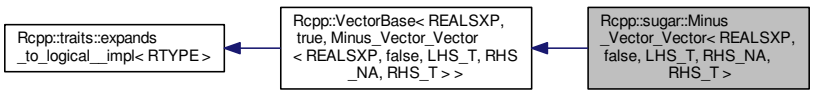
6.409 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.409.1 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.

6.409.2 Member Typedef Documentation

6.409.2.1 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.

6.409.2.2 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.

6.409.2.3 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.

6.409.2.4 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.

6.409.3 Constructor & Destructor Documentation

6.409.3.1 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.

6.409.4 Member Function Documentation

6.409.4.1 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.

6.409.4.2 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.

6.409.5 Member Data Documentation

6.409.5.1 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.

6.409.5.2 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.

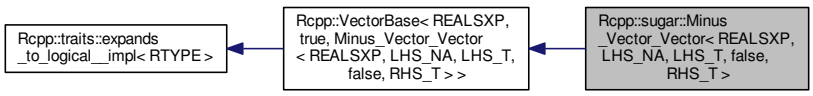
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

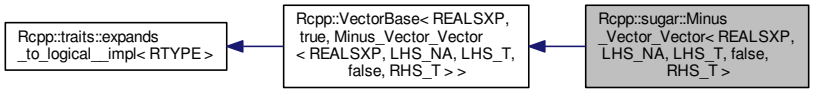
6.410 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.410.1 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.

6.410.2 Member Typedef Documentation

6.410.2.1 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.

6.410.2.2 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.

6.410.2.3 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.

6.410.2.4 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.

6.410.3 Constructor & Destructor Documentation

6.410.3.1 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.

6.410.4 Member Function Documentation

6.410.4.1 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.

6.410.4.2 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.

6.410.5 Member Data Documentation

6.410.5.1 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.

6.410.5.2 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.

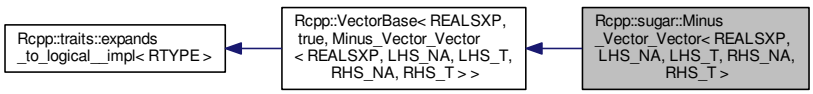
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

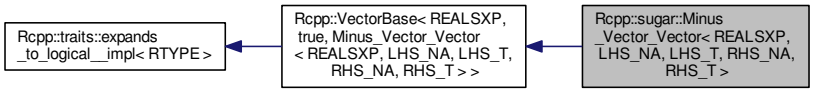
6.411 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.411.1 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.

6.411.2 Member Typedef Documentation

6.411.2.1 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.

6.411.2.2 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.

6.411.2.3 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.

6.411.2.4 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.

6.411.3 Constructor & Destructor Documentation

6.411.3.1 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.

6.411.4 Member Function Documentation

6.411.4.1 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.

6.411.4.2 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.

6.411.5 Member Data Documentation

6.411.5.1 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.

6.411.5.2 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.

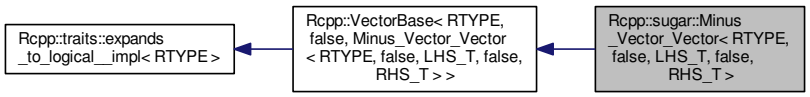
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

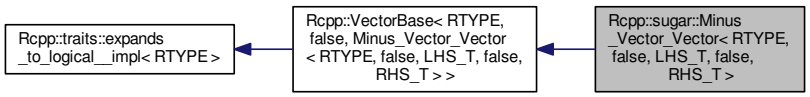
6.412 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.412.1 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.

6.412.2 Member Typedef Documentation

6.412.2.1 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.

6.412.2.2 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.

6.412.2.3 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.

6.412.2.4 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.

6.412.2.5 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.

6.412.3 Constructor & Destructor Documentation

6.412.3.1 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.

6.412.4 Member Function Documentation

6.412.4.1 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.

6.412.4.2 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.

6.412.5 Member Data Documentation

6.412.5.1 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.

6.412.5.2 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.

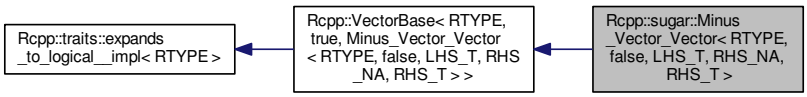
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

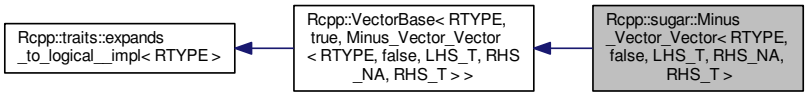
6.413 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.413.1 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.

6.413.2 Member Typedef Documentation

6.413.2.1 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.

6.413.2.2 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.

6.413.2.3 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.

6.413.2.4 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.

6.413.2.5 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.

6.413.3 Constructor & Destructor Documentation

6.413.3.1 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.

6.413.4 Member Function Documentation

6.413.4.1 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.

6.413.4.2 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.

6.413.5 Member Data Documentation

6.413.5.1 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.

6.413.5.2 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.

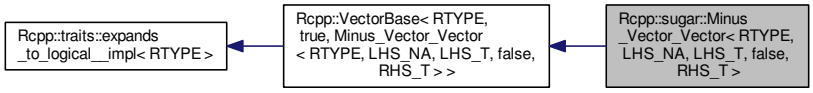
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

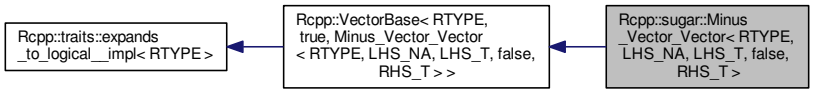
6.414 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.414.1 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.

6.414.2 Member Typedef Documentation

6.414.2.1 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.

6.414.2.2 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.

6.414.2.3 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.

6.414.2.4 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.

6.414.2.5 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.

6.414.3 Constructor & Destructor Documentation

6.414.3.1 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.

6.414.4 Member Function Documentation

6.414.4.1 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.

6.414.4.2 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.

6.414.5 Member Data Documentation

6.414.5.1 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.

6.414.5.2 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

6.415 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_, CppFunction *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

6.415.1 Detailed Description

holds information about exposed functions and classes

Definition at line 30 of file Module.h.

6.415.2 Member Typedef Documentation

6.415.2.1 typedef CLASS_MAP::iterator Rcpp::Module::CLASS_ITERATOR

Definition at line 37 of file Module.h.

6.415.2.2 typedef std::map<std::string,class_Base*> Rcpp::Module::CLASS_MAP

Definition at line 35 of file Module.h.

6.415.2.3 typedef std::pair<const std::string,class_Base*> Rcpp::Module::CLASS_PAIR

Definition at line 36 of file Module.h.

6.415.2.4 typedef std::pair<const std::string,CppFunction*> Rcpp::Module::FUNCTION_PAIR

Definition at line 33 of file Module.h.

6.415.2.5 typedef std::map<std::string,CppFunction*> Rcpp::Module::MAP

Definition at line 32 of file Module.h.

6.415.3 Constructor & Destructor Documentation

6.415.3.1 Rcpp::Module::Module ( ) [inline]

Definition at line 39 of file Module.h.

6.415.3.2 Rcpp::Module::Module (const char * *name_*) [inline]

Definition at line 42 of file Module.h.

References name, and prefix.

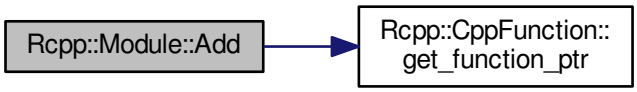
6.415.4 Member Function Documentation

6.415.4.1 void Rcpp::Module::Add (const char * *name_*, CppFunction * *ptr*) [inline]

Definition at line 193 of file Module.h.

References functions, Rcpp::CppFunction::get_function_ptr(), and prefix.

Here is the call graph for this function:



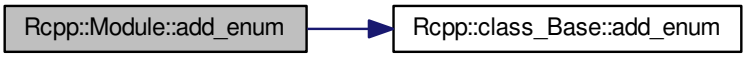
6.415.4.2 void Rcpp::Module::add_enum (const std::string & *parent_class_typeinfo_name*, const std::string & *enum_name*, const std::map< std::string, int > & *value*) [inline]

Definition at line 220 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:



6.415.4.3 void Rcpp::Module::AddClass (const char * *name_*, class_Base * *cptr*) [inline]

Definition at line 198 of file Module.h.

References classes.

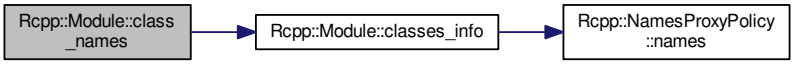
6.415.4.4 CharacterVector Rcpp::Module::class_names ( ) [inline]

exposed class names

Definition at line 102 of file Module.h.

References classes, and classes_info().

Here is the call graph for this function:



6.415.4.5 List Rcpp::Module::classes_info ( ) [inline]

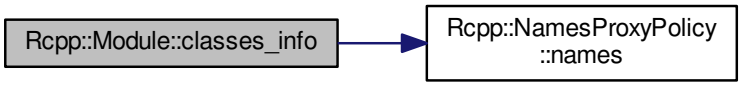
information about the classes

Definition at line 25 of file Module.h.

References classes, and Rcpp::NamesProxyPolicy< CLASS >::names().

Referenced by class_names().

Here is the call graph for this function:



6.415.4.6 CharacterVector Rcpp::Module::complete ( ) [inline]

completion information

Definition at line 120 of file Module.h.

References classes, and functions.

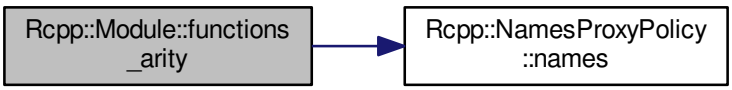
6.415.4.7 IntegerVector Rcpp::Module::functions_arity ( ) [inline]

vector of arity of all the functions exported by the module

Definition at line 73 of file Module.h.

References functions, and Rcpp::NamesProxyPolicy< CLASS >::names().

Here is the call graph for this function:



6.415.4.8 CharacterVector Rcpp::Module::functions_names ( ) [inline]

vector of names of the functions

Definition at line 89 of file Module.h.

References functions.

6.415.4.9 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.

Referenced by has_class().

6.415.4.10 class_Base* Rcpp::Module::get_class_pointer (const std::string & *cl*) [inline]

Definition at line 212 of file Module.h.

References classes.

6.415.4.11 SEXP Rcpp::Module::get_function (const std::string & *name_*) [inline]

Returns a list that contains:

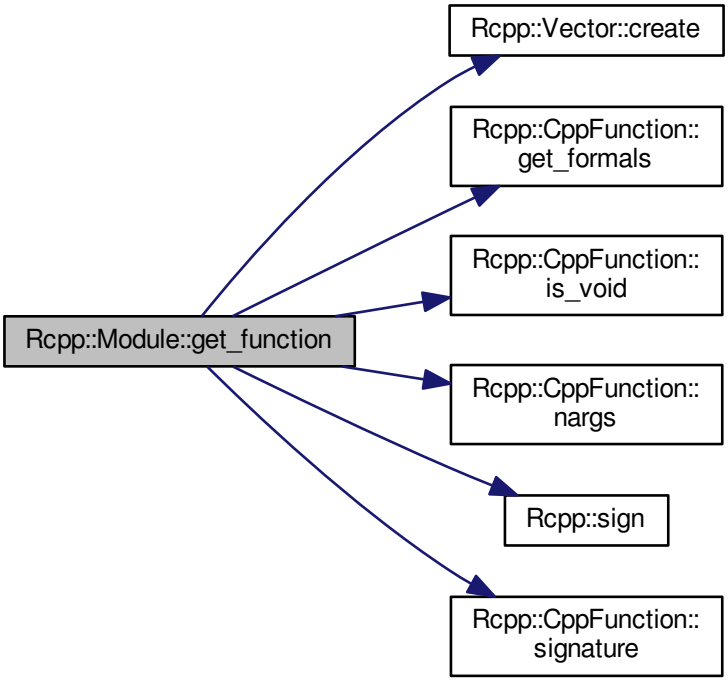
- an external pointer that encapsulates a CppFunction*
- voidness of the function (logical)
- docstring (character)
- signature (character)
- formal arguments of the function

The R code in Module.R uses this information to create a C++Function object

Definition at line 155 of file Module.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::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:



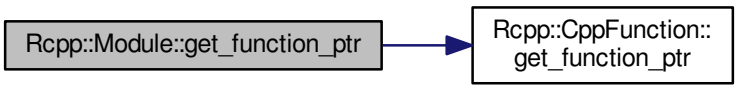
6.415.4.12 DL_FUNC Rcpp::Module::get_function_ptr (const std::string & *name_*) [inline]

get the underlying C++ function pointer as a DL_FUNC

Definition at line 180 of file Module.h.

References functions, and Rcpp::CppFunction::get_function_ptr().

Here is the call graph for this function:



6.415.4.13 bool Rcpp::Module::has_class (const std::string & *m*) [inline]

Definition at line 206 of file Module.h.

References classes, and get_class().

Here is the call graph for this function:

6.415.4.14 bool Rcpp::Module::has_function (const std::string & *m*) [inline]

Definition at line 202 of file Module.h.

References functions.

6.415.4.15 SEXP Rcpp::Module::invoke (const std::string & *name_*, SEXP * *args*, int *nargs*) [inline]

calls a function from that module with the specified arguments

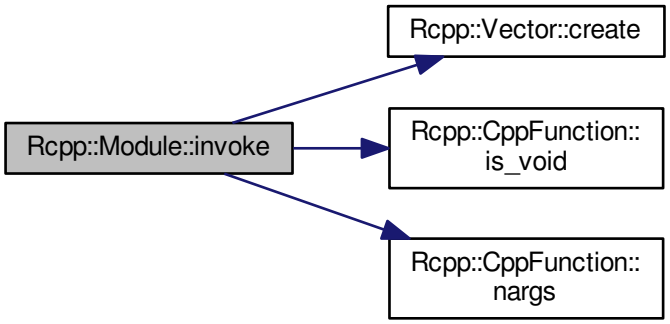
Parameters

name	the name of the function to call
args	an array of R objects to use as arguments for the function
nargs	number of arguments

Definition at line 54 of file Module.h.

References Rcpp::_, Rcpp::Vector< RTYPE, StoragePolicy >::create(), functions, Rcpp::CppFunction::is_void(), and Rcpp::CppFunction::nargs().

Here is the call graph for this function:



6.415.5 Member Data Documentation

6.415.5.1 CLASS_MAP Rcpp::Module::classes [private]

Definition at line 236 of file Module.h.

Referenced by add_enum(), AddClass(), class_names(), classes_info(), complete(), get_class(), get_class_pointer(), and has_class().

6.415.5.2 MAP Rcpp::Module::functions [private]

Definition at line 235 of file Module.h.

Referenced by Add(), complete(), functions_arity(), functions_names(), get_function(), get_function_ptr(), has_↩ function(), and invoke().

6.415.5.3 std::string Rcpp::Module::name

Definition at line 218 of file Module.h.

Referenced by Module().

6.415.5.4 std::string Rcpp::Module::prefix [private]

Definition at line 237 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

6.416 Rcpp::traits::module_wrap_traits< T > Struct Template Reference

`#include <module_wrap_traits.h>`

Public Types

- typedef normal_wrap_tag category

6.416.1 Detailed Description

template<typename T>
struct Rcpp::traits::module_wrap_traits< T >

Definition at line 33 of file module_wrap_traits.h.

6.416.2 Member Typedef Documentation

6.416.2.1 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

6.417 Rcpp::traits::module_wrap_traits< T * > Struct Template Reference

`#include <module_wrap_traits.h>`

Public Types

- typedef pointer_wrap_tag category

6.417.1 Detailed Description

template<typename T>
struct Rcpp::traits::module_wrap_traits< T * >

Definition at line 35 of file module_wrap_traits.h.

6.417.2 Member Typedef Documentation

6.417.2.1 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

6.418 Rcpp::traits::module_wrap_traits< void > Struct Template Reference

`#include <module_wrap_traits.h>`

Public Types

- typedef void_wrap_tag category

6.418.1 Detailed Description

template<>
struct Rcpp::traits::module_wrap_traits< void >

Definition at line 34 of file module_wrap_traits.h.

6.418.2 Member Typedef Documentation

6.418.2.1 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

6.419 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)

6.419.1 Detailed Description

Definition at line 22 of file Na_Proxy.h.

6.419.2 Friends And Related Function Documentation

6.419.2.1 bool operator== (double *x*, Rcpp::Na_Proxy) [friend]

Definition at line 24 of file Na_Proxy.h.

6.419.2.2 bool operator== (int *x*, Rcpp::Na_Proxy) [friend]

Definition at line 25 of file Na_Proxy.h.

6.419.2.3 bool operator== (Rcpp::String *x*, Rcpp::Na_Proxy) [friend]

Definition at line 26 of file Na_Proxy.h.

6.419.2.4 bool operator== (Rcomplex *x*, Rcpp::Na_Proxy) [friend]

Definition at line 27 of file Na_Proxy.h.

6.419.2.5 bool operator== (SEXP *x*, Rcpp::Na_Proxy) [friend]

Definition at line 28 of file Na_Proxy.h.

6.419.2.6 bool operator== (std::string , Rcpp::Na_Proxy) [friend]

Definition at line 29 of file Na_Proxy.h.

6.419.2.7 bool operator== (const char * , Rcpp::Na_Proxy) [friend]

Definition at line 30 of file Na_Proxy.h.

6.419.2.8 bool operator== (Rcpp::internal::string_proxy< STRSXP > *x*, Rcpp::Na_Proxy) [friend]

Definition at line 31 of file Na_Proxy.h.

6.419.2.9 bool operator== (Rcpp::internal::const_string_proxy< STRSXP > *x*, Rcpp::Na_Proxy) [friend]

Definition at line 34 of file Na_Proxy.h.

6.419.2.10 bool operator== (Rcpp::Na_Proxy , double *x*) [friend]

Definition at line 38 of file Na_Proxy.h.

6.419.2.11 bool operator== (Rcpp::Na_Proxy , int *x*) [friend]

Definition at line 39 of file Na_Proxy.h.

6.419.2.12 bool operator== (Rcpp::Na_Proxy , Rcpp::String *x*) [friend]

Definition at line 40 of file Na_Proxy.h.

6.419.2.13 bool operator== (Rcpp::Na_Proxy , SEXP *x*) [friend]

Definition at line 41 of file Na_Proxy.h.

6.419.2.14 bool operator== (Rcpp::Na_Proxy , Rcomplex *x*) [friend]

Definition at line 42 of file Na_Proxy.h.

6.419.2.15 bool operator== (Rcpp::Na_Proxy , std::string) [friend]

Definition at line 43 of file Na_Proxy.h.

6.419.2.16 bool operator== (Rcpp::Na_Proxy , const char *) [friend]

Definition at line 44 of file Na_Proxy.h.

6.419.2.17 bool operator== (Rcpp::Na_Proxy , Rcpp::internal::string_proxy< STRSXP > *x*) [friend]

Definition at line 45 of file Na_Proxy.h.

6.419.2.18 bool operator== (Rcpp::Na_Proxy , Rcpp::internal::const_string_proxy< STRSXP > *x*) [friend]

Definition at line 48 of file Na_Proxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Na_Proxy.h

6.420 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

6.420.1 Detailed Description

template<typename T>
class Rcpp::traits::named_object< T >

Definition at line 34 of file named_object.h.

6.420.2 Constructor & Destructor Documentation

6.420.2.1 template<typename T> Rcpp::traits::named_object< T >::named_object (const std::string & *name_*, const T & *o_*) [inline]

Definition at line 36 of file named_object.h.

6.420.3 Member Data Documentation

6.420.3.1 template<typename T> const std::string& Rcpp::traits::named_object< T >::name

Definition at line 38 of file named_object.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

6.420.3.2 template<typename T> const T& Rcpp::traits::named_object< T >::object

Definition at line 39 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

6.421 Rcpp::traits::named_object< SEXP > Class Template 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

6.421.1 Detailed Description

template<>
class Rcpp::traits::named_object< SEXP >

Definition at line 41 of file named_object.h.

6.421.2 Constructor & Destructor Documentation

6.421.2.1 Rcpp::traits::named_object< SEXP >::named_object (const std::string & *name_*, const SEXP & *o_*) [inline]

Definition at line 43 of file named_object.h.

6.421.2.2 Rcpp::traits::named_object< SEXP >::named_object (const named_object< SEXP > & *other*) [inline]

Definition at line 49 of file named_object.h.

6.421.2.3 Rcpp::traits::named_object< SEXP >::~named_object ( ) [inline]

Definition at line 54 of file named_object.h.

6.421.3 Member Data Documentation

6.421.3.1 const std::string& Rcpp::traits::named_object< SEXP >::name

Definition at line 57 of file named_object.h.

6.421.3.2 SEXP Rcpp::traits::named_object< SEXP >::object

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

6.422 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

6.422.1 Detailed Description

Definition at line 50 of file Named.h.

6.422.2 Constructor & Destructor Documentation

6.422.2.1 Rcpp::internal::NamedPlaceHolder::NamedPlaceHolder ( ) [inline]

Definition at line 52 of file Named.h.

6.422.2.2 Rcpp::internal::NamedPlaceHolder::~NamedPlaceHolder ( ) [inline]

Definition at line 53 of file Named.h.

6.422.3 Member Function Documentation

6.422.3.1 Rcpp::internal::NamedPlaceHolder::operator SEXP ( ) const [inline]

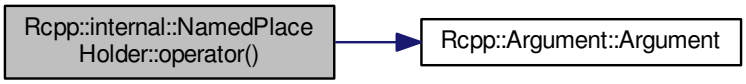
Definition at line 60 of file Named.h.

6.422.3.2 Argument Rcpp::internal::NamedPlaceHolder::operator() (const std::string & *arg*) const [inline]

Definition at line 57 of file Named.h.

References Rcpp::Argument::Argument().

Here is the call graph for this function:

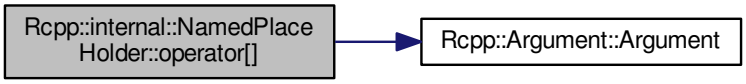


6.422.3.3 Argument Rcpp::internal::NamedPlaceHolder::operator[ ] (const std::string & *arg*) const [inline]

Definition at line 54 of file Named.h.

References Rcpp::Argument::Argument().

Here is the call graph for this function:



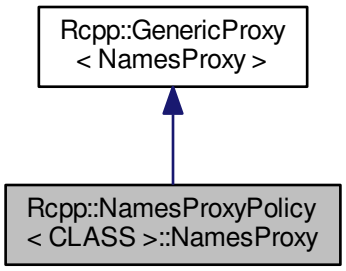
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Named.h

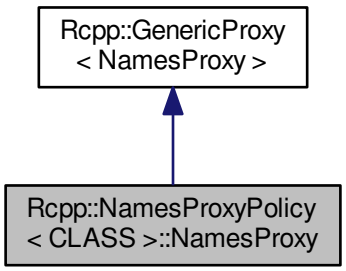
6.423 Rcpp::NamesProxyPolicy< CLASS >::NamesProxy Class Reference

`#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

6.423.1 Detailed Description

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >::NamesProxy

Definition at line 27 of file NamesProxy.h.

6.423.2 Constructor & Destructor Documentation

6.423.2.1 template<typename CLASS> Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::NamesProxy (CLASS & *v*) [inline]

Definition at line 29 of file NamesProxy.h.

Referenced by Rcpp::NamesProxyPolicy< Vector< RTYPE, StoragePolicy > >::names().

6.423.3 Member Function Documentation

6.423.3.1 template<typename CLASS> SEXP Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get ( ) const [inline], [private]

Definition at line 45 of file NamesProxy.h.

References RCPP_GET_NAMES.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=().

6.423.3.2 template<typename CLASS > template<typename T > Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator T ( ) const

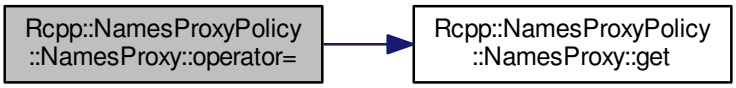
Definition at line 70 of file proxy.h.

6.423.3.3 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().

Here is the call graph for this function:



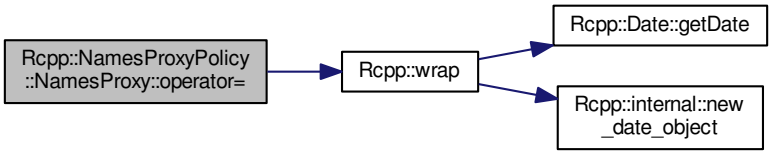
6.423.3.4 template<typename CLASS> template<typename T > NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (const T & *rhs*)

6.423.3.5 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:



6.423.3.6 template<typename CLASS> void Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set (SEXP *x*) [inline], [private]

Definition at line 49 of file NamesProxy.h.

References Rcpp::Rcpp_eval().

Here is the call graph for this function:

6.423.4 Member Data Documentation

6.423.4.1 template<typename CLASS> CLASS& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent [private]

Definition at line 43 of file NamesProxy.h.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/NamesProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.424 Rcpp::NamesProxyPolicy< CLASS > Class Template Reference

`#include <NamesProxy.h>`

Classes

- class const_NamesProxy
- class NamesProxy

Public Member Functions

- NamesProxy names ()
- const_NamesProxy names () const

6.424.1 Detailed Description

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >

Definition at line 24 of file NamesProxy.h.

6.424.2 Member Function Documentation

6.424.2.1 template<typename CLASS> NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( ) [inline]

Definition at line 80 of file NamesProxy.h.

Referenced by Rcpp::Module::classes_info(), class_< Class >::fields(), Rcpp::Module::functions_arity(), class_< Class >::getMethods(), lapplyCpp(), class_< Class >::methods_arity(), class_< Class >::methods_voidness(), and class_< Class >::property_classes().

6.424.2.2 template<typename CLASS> const_NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( ) const [inline]

Definition at line 84 of file NamesProxy.h.

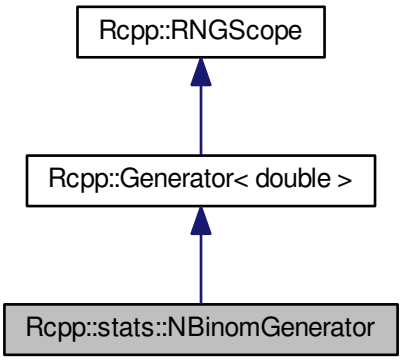
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/NamesProxy.h

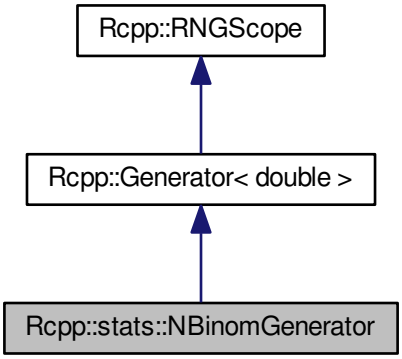
6.425 Rcpp::stats::NBinomGenerator Class Reference

`#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

6.425.1 Detailed Description

Definition at line 28 of file rnbinom.h.

6.425.2 Constructor & Destructor Documentation

6.425.2.1 Rcpp::stats::NBinomGenerator::NBinomGenerator (double *siz_*, double *prob_*) [inline]

Definition at line 31 of file rnbinom.h.

6.425.3 Member Function Documentation

6.425.3.1 double Rcpp::stats::NBinomGenerator::operator() ( ) const [inline]

Definition at line 34 of file rnbinom.h.

References lambda, and siz.

6.425.4 Member Data Documentation

6.425.4.1 double Rcpp::stats::NBinomGenerator::lambda [private]

Definition at line 40 of file rnbinom.h.

Referenced by operator()().

6.425.4.2 double Rcpp::stats::NBinomGenerator::siz [private]

Definition at line 39 of file rnbinom.h.

Referenced by operator()().

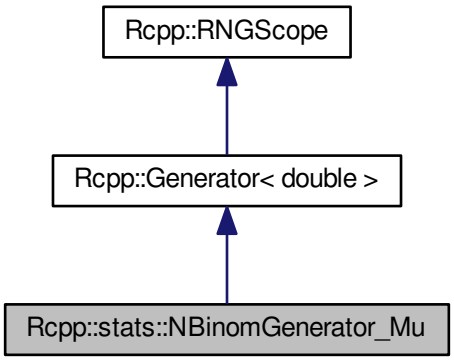
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnbinom.h

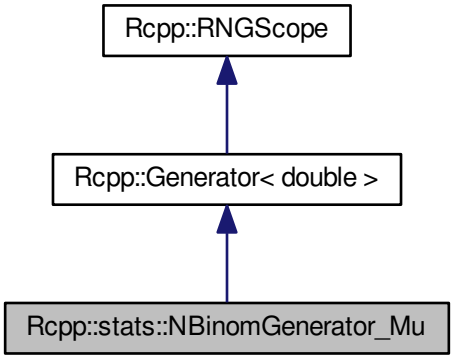
6.426 Rcpp::stats::NBinomGenerator_Mu Class Reference

`#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

6.426.1 Detailed Description

Definition at line 28 of file rnbinom_mu.h.

6.426.2 Constructor & Destructor Documentation

6.426.2.1 Rcpp::stats::NBinomGenerator_Mu::NBinomGenerator_Mu (double *siz_*, double *mu_*) [inline]

Definition at line 31 of file rnbinom_mu.h.

6.426.3 Member Function Documentation

6.426.3.1 double Rcpp::stats::NBinomGenerator_Mu::operator() ( ) const [inline]

Definition at line 34 of file rnbinom_mu.h.

References lambda, and siz.

6.426.4 Member Data Documentation

6.426.4.1 double Rcpp::stats::NBinomGenerator_Mu::lambda [private]

Definition at line 40 of file rnbinom_mu.h.

Referenced by operator()().

6.426.4.2 double Rcpp::stats::NBinomGenerator_Mu::siz [private]

Definition at line 39 of file rnbinom_mu.h.

Referenced by operator()().

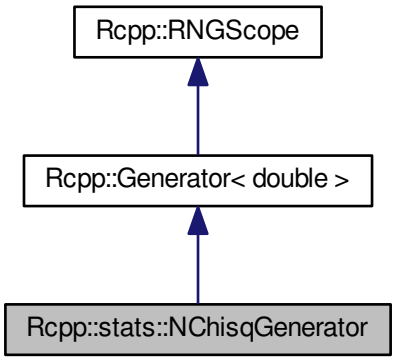
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnbinom_mu.h

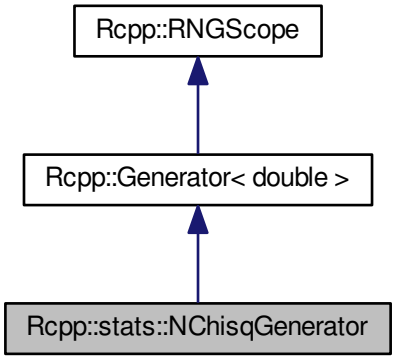
6.427 Rcpp::stats::NChisqGenerator Class Reference

`#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

6.427.1 Detailed Description

Definition at line 28 of file rnchisq.h.

6.427.2 Constructor & Destructor Documentation

6.427.2.1 Rcpp::stats::NChisqGenerator::NChisqGenerator (double *df_*, double *lambda_*) [inline]

Definition at line 31 of file rnchisq.h.

6.427.3 Member Function Documentation

6.427.3.1 double Rcpp::stats::NChisqGenerator::operator() ( ) const [inline]

Definition at line 34 of file rnchisq.h.

References df, df_2, and lambda_2.

6.427.4 Member Data Documentation

6.427.4.1 double Rcpp::stats::NChisqGenerator::df [private]

Definition at line 46 of file rnchisq.h.

Referenced by operator()().

6.427.4.2 double Rcpp::stats::NChisqGenerator::df_2 [private]

Definition at line 47 of file rnchisq.h.

Referenced by operator()().

6.427.4.3 double Rcpp::stats::NChisqGenerator::lambda_2 [private]

Definition at line 48 of file rnchisq.h.

Referenced by operator()().

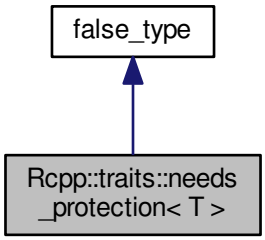
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnchisq.h

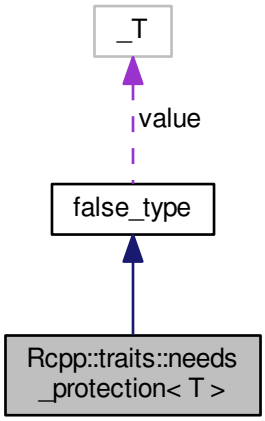
6.428 Rcpp::traits::needs_protection< T > Struct Template Reference

`#include <named_object.h>`

Inheritance diagram for Rcpp::traits::needs_protection< T >:



Collaboration diagram for Rcpp::traits::needs_protection< T >:



Additional Inherited Members

6.428.1 Detailed Description

template<typename T>
struct Rcpp::traits::needs_protection< T >

Definition at line 31 of file named_object.h.

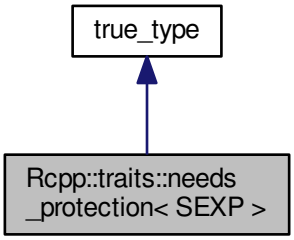
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

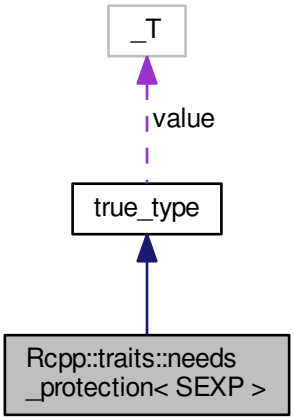
6.429 Rcpp::traits::needs_protection< SEXP > Struct Template Reference

`#include <named_object.h>`

Inheritance diagram for Rcpp::traits::needs_protection< SEXP >:



Collaboration diagram for Rcpp::traits::needs_protection< SEXP >:



Additional Inherited Members

6.429.1 Detailed Description

template<>
struct Rcpp::traits::needs_protection< SEXP >

Definition at line 32 of file named_object.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

6.430 Rcpp::sugar::negate< NA > Struct Template Reference

`#include <not.h>`

Static Public Member Functions

- static int apply (int x)

6.430.1 Detailed Description

template<bool NA>
struct Rcpp::sugar::negate< NA >

Definition at line 28 of file not.h.

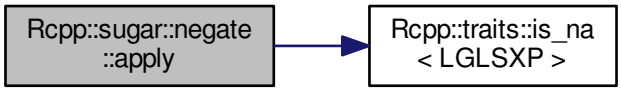
6.430.2 Member Function Documentation

6.430.2.1 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

6.431 Rcpp::sugar::negate< false > Struct Template Reference

`#include <not.h>`

Static Public Member Functions

- static int apply (int x)

6.431.1 Detailed Description

template<>
struct Rcpp::sugar::negate< false >

Definition at line 34 of file not.h.

6.431.2 Member Function Documentation

6.431.2.1 static int Rcpp::sugar::negate< false >::apply (int *x*) [inline], [static]

Definition at line 35 of file not.h.

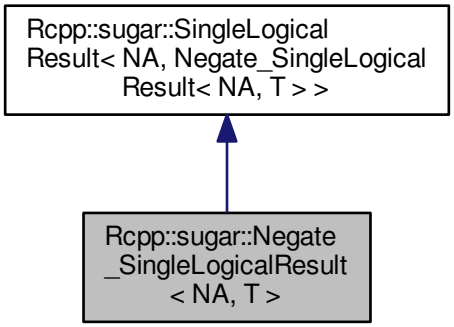
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/not.h

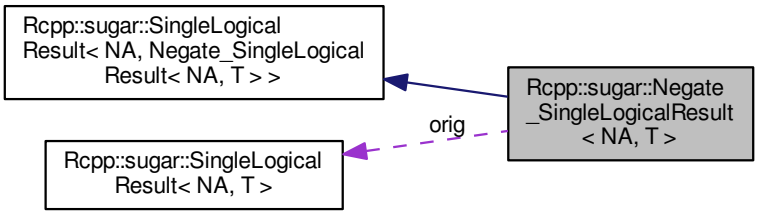
6.432 Rcpp::sugar::Negate_SingleLogicalResult< NA, T > Class Template Reference

`#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

6.432.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Negate_SingleLogicalResult< NA, T >

Definition at line 42 of file not.h.

6.432.2 Member Typedef Documentation

6.432.2.1 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.

6.432.2.2 template<bool NA, typename T > typedef SingleLogicalResult<NA,T> Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::TYPE

Definition at line 44 of file not.h.

6.432.3 Constructor & Destructor Documentation

6.432.3.1 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.

6.432.4 Member Function Documentation

6.432.4.1 template<bool NA, typename T > void Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply ( ) [inline]

Definition at line 48 of file not.h.

6.432.5 Member Data Documentation

6.432.5.1 template<bool NA, typename T > const TYPE& Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::orig [private]

Definition at line 53 of file not.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/not.h

6.433 Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no Struct Reference

Public Attributes

- char array [2]

6.433.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no

Definition at line 376 of file cbind.h.

6.433.2 Member Data Documentation

6.433.2.1 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

6.434 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

6.434.1 Detailed Description

Definition at line 47 of file no_init.h.

6.434.2 Constructor & Destructor Documentation

6.434.2.1 Rcpp::no_init_matrix::no_init_matrix (int *nr_*, int *nc_*) [inline]

Definition at line 49 of file no_init.h.

6.434.3 Member Function Documentation

6.434.3.1 int Rcpp::no_init_matrix::ncol ( ) const [inline]

Definition at line 55 of file no_init.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix().

6.434.3.2 int Rcpp::no_init_matrix::nrow ( ) const [inline]

Definition at line 51 of file no_init.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix().

6.434.3.3 template<int RTYPE, template< class > class StoragePolicy> Rcpp::no_init_matrix::operator Matrix< RTYPE, StoragePolicy > ( ) const [inline]

Definition at line 60 of file no_init.h.

6.434.4 Member Data Documentation

6.434.4.1 int Rcpp::no_init_matrix::nc [private]

Definition at line 66 of file no_init.h.

6.434.4.2 int Rcpp::no_init_matrix::nr [private]

Definition at line 65 of file no_init.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/no_init.h

6.435 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

6.435.1 Detailed Description

Definition at line 30 of file no_init.h.

6.435.2 Constructor & Destructor Documentation

6.435.2.1 Rcpp::no_init_vector::no_init_vector (R_xlen_t *size_*) [inline]

Definition at line 32 of file no_init.h.

Referenced by Rcpp::no_init().

6.435.3 Member Function Documentation

6.435.3.1 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< INTSXP >::Vector().

6.435.3.2 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.

6.435.4 Member Data Documentation

6.435.4.1 R_xlen_t Rcpp::no_init_vector::size [private]

Definition at line 44 of file no_init.h.

Referenced by get(), and operator Vector< RTYPE, StoragePolicy >().

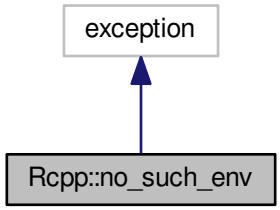
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/no_init.h

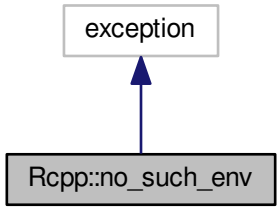
6.436 Rcpp::no_such_env Class Reference

`#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

6.436.1 Detailed Description

Definition at line 50 of file exceptions.h.

6.436.2 Constructor & Destructor Documentation

6.436.2.1 Rcpp::no_such_env::no_such_env (const std::string & *name*) throw) [inline]

Definition at line 52 of file exceptions.h.

6.436.2.2 Rcpp::no_such_env::no_such_env (int *pos*) throw) [inline]

Definition at line 53 of file exceptions.h.

6.436.2.3 virtual Rcpp::no_such_env::~no_such_env ( ) throw) [inline], [virtual]

Definition at line 54 of file exceptions.h.

6.436.3 Member Function Documentation

6.436.3.1 virtual const char* Rcpp::no_such_env::what ( ) const throw) [inline], [virtual]

Definition at line 55 of file exceptions.h.

References Rcpp::exception::message.

6.436.4 Member Data Documentation

6.436.4.1 std::string Rcpp::no_such_env::message [private]

Definition at line 55 of file exceptions.h.

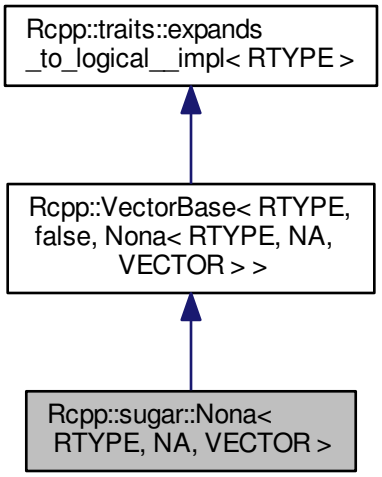
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

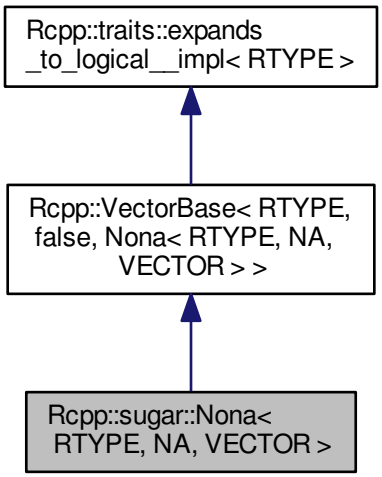
6.437 Rcpp::sugar::Nona< RTYPE, NA, VECTOR > Class Template Reference

`#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

6.437.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
class Rcpp::sugar::Nona< RTYPE, NA, VECTOR >

Definition at line 29 of file nona.h.

6.437.2 Member Typedef Documentation

6.437.2.1 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.

6.437.2.2 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.

6.437.3 Constructor & Destructor Documentation

6.437.3.1 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.

6.437.4 Member Function Documentation

6.437.4.1 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.

6.437.4.2 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.

6.437.5 Member Data Documentation

6.437.5.1 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().

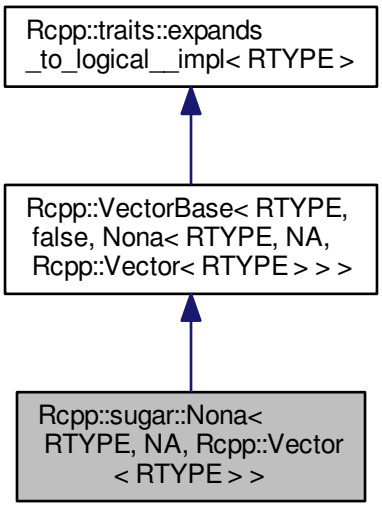
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

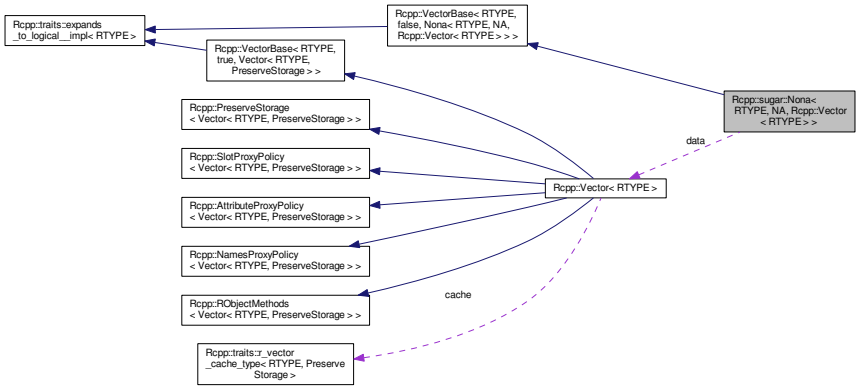
6.438 Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > Class Template Reference

```
#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

6.438.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >

Definition at line 45 of file nona.h.

6.438.2 Member Typedef Documentation

6.438.2.1 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.

6.438.2.2 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.

6.438.2.3 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.

6.438.3 Constructor & Destructor Documentation

6.438.3.1 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.

6.438.4 Member Function Documentation

6.438.4.1 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.

6.438.4.2 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.

6.438.5 Member Data Documentation

6.438.5.1 template<int RTYPE, bool NA> iterator Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::data [private]

Definition at line 57 of file nona.h.

6.438.5.2 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

6.439 Rcpp::sugar::NonaPrimitive< T > Class Template Reference

`#include <nona.h>`

Public Member Functions

- NonaPrimitive (T t)
- operator T ()

Private Attributes

- T x

6.439.1 Detailed Description

template<typename T>
class Rcpp::sugar::NonaPrimitive< T >

Definition at line 62 of file nona.h.

6.439.2 Constructor & Destructor Documentation

6.439.2.1 template<typename T> Rcpp::sugar::NonaPrimitive< T >::NonaPrimitive (T *t*) [inline]

Definition at line 64 of file nona.h.

6.439.3 Member Function Documentation

6.439.3.1 template<typename T> Rcpp::sugar::NonaPrimitive< T >::operator T ( ) [inline]

Definition at line 65 of file nona.h.

6.439.4 Member Data Documentation

6.439.4.1 template<typename T> T Rcpp::sugar::NonaPrimitive< T >::x [private]

Definition at line 68 of file nona.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

6.440 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

6.440.1 Detailed Description

template<typename CLASS>
class Rcpp::NoProtectStorage< CLASS >

Definition at line 7 of file NoProtectStorage.h.

6.440.2 Constructor & Destructor Documentation

6.440.2.1 template<typename CLASS > Rcpp::NoProtectStorage< CLASS >::NoProtectStorage ( ) [inline]

Definition at line 10 of file NoProtectStorage.h.

6.440.2.2 template<typename CLASS > Rcpp::NoProtectStorage< CLASS >::~NoProtectStorage ( ) [inline]

Definition at line 12 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

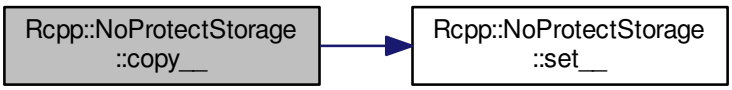
6.440.3 Member Function Documentation

6.440.3.1 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:



6.440.3.2 template<typename CLASS > SEXP Rcpp::NoProtectStorage< CLASS >::get__ ( ) const [inline]

Definition at line 24 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.440.3.3 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.

6.440.3.4 template<typename CLASS > SEXP Rcpp::NoProtectStorage< CLASS >::invalidate__ ( ) [inline]

Definition at line 28 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.440.3.5 template<typename CLASS > Rcpp::NoProtectStorage< CLASS >::operator SEXP ( ) const [inline]

Definition at line 44 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.440.3.6 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__().

6.440.4 Member Data Documentation

6.440.4.1 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

6.441 Rcpp::traits::normal_wrap_tag Struct Reference

`#include <module_wrap_traits.h>`

6.441.1 Detailed Description

Definition at line 29 of file module_wrap_traits.h.

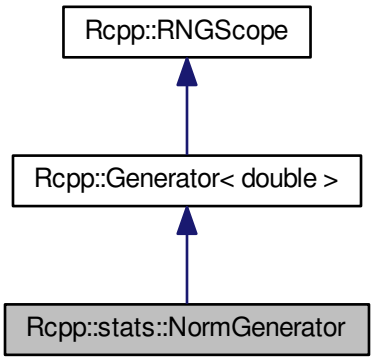
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

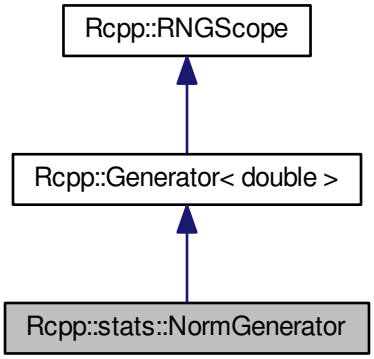
6.442 Rcpp::stats::NormGenerator Class Reference

`#include <rnorm.h>`

Inheritance diagram for Rcpp::stats::NormGenerator:



Collaboration diagram for Rcpp::stats::NormGenerator:



Public Member Functions

- NormGenerator (double mean_=0.0, double sd_=1.0)
- double operator() () const

Private Attributes

- double mean
- double sd

Additional Inherited Members

6.442.1 Detailed Description

Definition at line 28 of file rnorm.h.

6.442.2 Constructor & Destructor Documentation

6.442.2.1 Rcpp::stats::NormGenerator::NormGenerator (double *mean_* = 0.0, double *sd_* = 1.0) [inline]

Definition at line 31 of file rnorm.h.

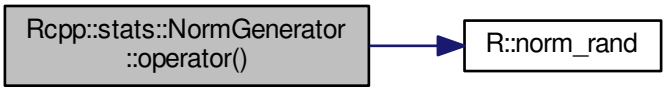
6.442.3 Member Function Documentation

6.442.3.1 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:



6.442.4 Member Data Documentation

6.442.4.1 double Rcpp::stats::NormGenerator::mean [private]

Definition at line 39 of file rnorm.h.

Referenced by operator()(), and Rcpp::stats::NormGenerator__sd1::operator()().

6.442.4.2 double Rcpp::stats::NormGenerator::sd [private]

Definition at line 40 of file rnorm.h.

Referenced by operator()(), and Rcpp::stats::NormGenerator__mean0::operator()().

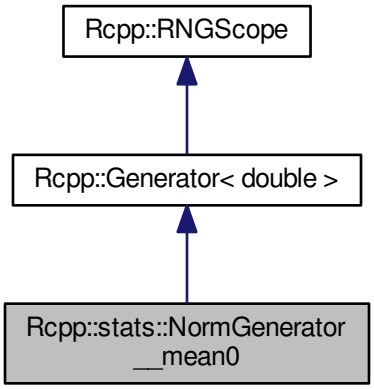
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

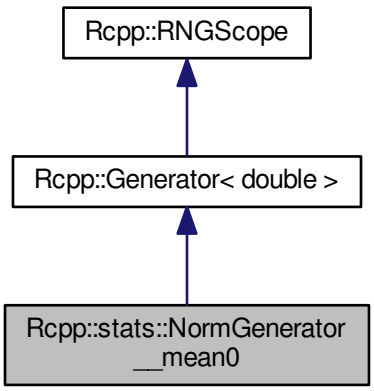
6.443 Rcpp::stats::NormGenerator__mean0 Class Reference

`#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

6.443.1 Detailed Description

Definition at line 56 of file rnorm.h.

6.443.2 Constructor & Destructor Documentation

6.443.2.1 Rcpp::stats::NormGenerator__mean0::NormGenerator__mean0 (double *sd_* = 1.0) [inline]

Definition at line 59 of file rnorm.h.

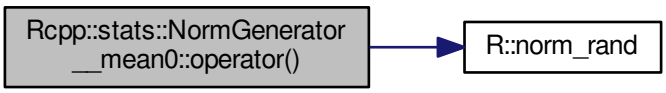
6.443.3 Member Function Documentation

6.443.3.1 double Rcpp::stats::NormGenerator__mean0::operator() ( ) const [inline]

Definition at line 61 of file rnorm.h.

References R::norm_rand(), and Rcpp::stats::NormGenerator::sd.

Here is the call graph for this function:



6.443.4 Member Data Documentation

6.443.4.1 double Rcpp::stats::NormGenerator__mean0::sd [private]

Definition at line 66 of file rnorm.h.

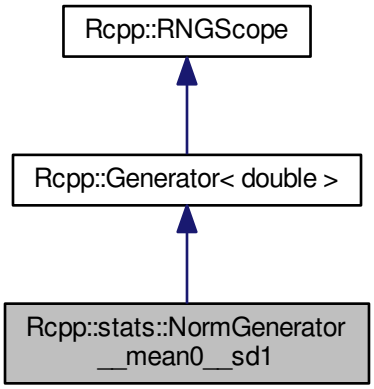
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

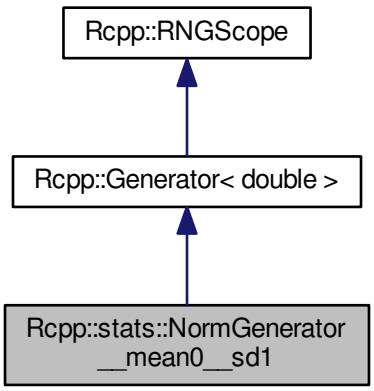
6.444 Rcpp::stats::NormGenerator__mean0__sd1 Class Reference

`#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

6.444.1 Detailed Description

Definition at line 69 of file rnorm.h.

6.444.2 Constructor & Destructor Documentation

6.444.2.1 Rcpp::stats::NormGenerator__mean0__sd1::NormGenerator__mean0__sd1 ( ) [inline]

Definition at line 72 of file rnorm.h.

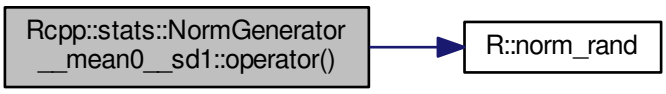
6.444.3 Member Function Documentation

6.444.3.1 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:



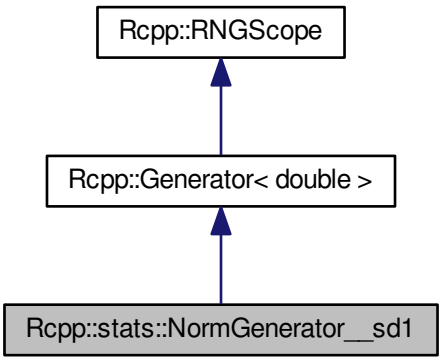
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

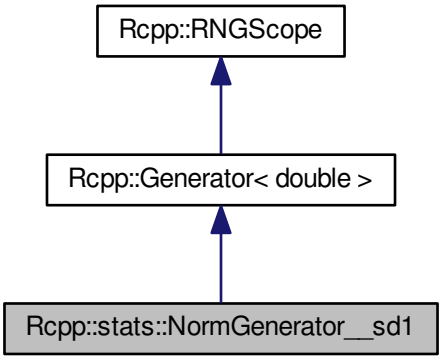
6.445 Rcpp::stats::NormGenerator__sd1 Class Reference

`#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

6.445.1 Detailed Description

Definition at line 43 of file rnorm.h.

6.445.2 Constructor & Destructor Documentation

6.445.2.1 Rcpp::stats::NormGenerator__sd1::NormGenerator__sd1 (double *mean_* = 0.0) [inline]

Definition at line 46 of file rnorm.h.

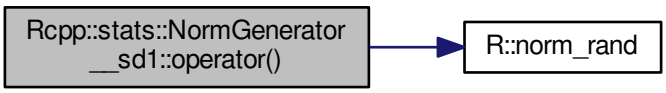
6.445.3 Member Function Documentation

6.445.3.1 double Rcpp::stats::NormGenerator__sd1::operator() ( ) const [inline]

Definition at line 48 of file rnorm.h.

References Rcpp::stats::NormGenerator::mean, and R::norm_rand().

Here is the call graph for this function:



6.445.4 Member Data Documentation

6.445.4.1 double Rcpp::stats::NormGenerator__sd1::mean [private]

Definition at line 53 of file rnorm.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

6.446 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

6.446.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::not_< RTYPE, NA >

Definition at line 29 of file not.h.

6.446.2 Member Typedef Documentation

6.446.2.1 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.

6.446.3 Member Function Documentation

6.446.3.1 template<int RTYPE, bool NA> int Rcpp::sugar::not_< RTYPE, NA >::apply (STORAGE *x*) const [inline]

Definition at line 32 of file not.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.447 Rcpp::sugar::not_< CPLXSXP, false > Class Template Reference

`#include <not.h>`

Public Member Functions

- int apply (Rcomplex x) const

6.447.1 Detailed Description

template<>
class Rcpp::sugar::not_< CPLXSXP, false >

Definition at line 66 of file not.h.

6.447.2 Member Function Documentation

6.447.2.1 int Rcpp::sugar::not_< CPLXSXP, 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

6.448 Rcpp::sugar::not_< CPLXSXP, NA > Class Template Reference

```
#include <not.h>
```

Public Member Functions

- int apply (Rcomplex x) const

6.448.1 Detailed Description

template<bool NA>
class Rcpp::sugar::not_< CPLXSXP, NA >

Definition at line 59 of file not.h.

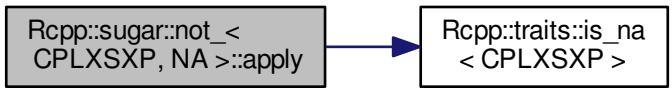
6.448.2 Member Function Documentation

6.448.2.1 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

6.449 Rcpp::sugar::not_< REALSXP, false > Class Template Reference

`#include <not.h>`

Public Member Functions

- int apply (double x) const

6.449.1 Detailed Description

template<>
class Rcpp::sugar::not_< REALSXP, false >

Definition at line 52 of file not.h.

6.449.2 Member Function Documentation

6.449.2.1 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

6.450 Rcpp::sugar::not_< REALSXP, NA > Class Template Reference

`#include <not.h>`

Public Member Functions

- int apply (double x) const

6.450.1 Detailed Description

template<bool NA>
class Rcpp::sugar::not_< REALSXP, NA >

Definition at line 45 of file not.h.

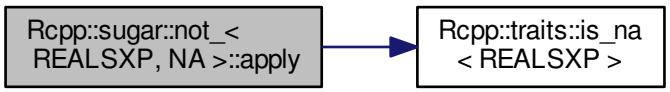
6.450.2 Member Function Documentation

6.450.2.1 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

6.451 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

6.451.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::not_< RTYPE, false >

Definition at line 37 of file not.h.

6.451.2 Member Typedef Documentation

6.451.2.1 template<int RTYPE> typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::not_< RTYPE, false >::STORAGE

Definition at line 39 of file not.h.

6.451.3 Member Function Documentation

6.451.3.1 template<int RTYPE> int Rcpp::sugar::not_< RTYPE, false >::apply (STORAGE *x*) const [inline]

Definition at line 40 of file not.h.

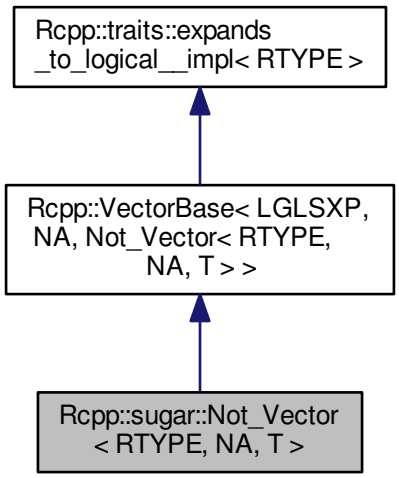
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

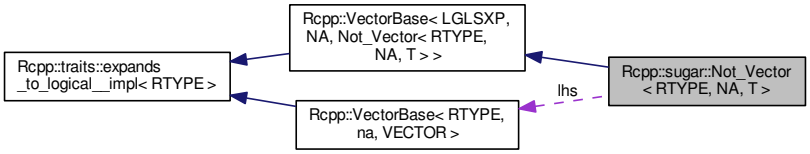
6.452 Rcpp::sugar::Not_Vector< RTYPE, NA, T > Class Template Reference

`#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

6.452.1 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.

6.452.2 Member Typedef Documentation

6.452.2.1 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.

6.452.2.2 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.

6.452.2.3 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.

6.452.3 Constructor & Destructor Documentation

6.452.3.1 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.

6.452.4 Member Function Documentation

6.452.4.1 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.

6.452.4.2 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.

6.452.5 Member Data Documentation

6.452.5.1 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.

6.452.5.2 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.453 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

6.453.1 Detailed Description

template<class T>
class Rcpp::Nullable< T >

Definition at line 33 of file Nullable.h.

6.453.2 Constructor & Destructor Documentation

6.453.2.1 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.

6.453.2.2 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.

6.453.2.3 template<class T > Rcpp::Nullable< T >::Nullable (SEXP *t*) [inline]

Standard constructor of a Nullable object

Parameters

SEXP	is stored

Definition at line 56 of file Nullable.h.

References Rcpp::Nullable< T >::m_set, and Rcpp::Nullable< T >::m_sexp.

6.453.3 Member Function Documentation

6.453.3.1 template<class T > T Rcpp::Nullable< T >::as ( ) [inline]

Returns m_sexp as a T

Definition at line 129 of file Nullable.h.

6.453.3.2 template<class T > void Rcpp::Nullable< T >::checkIfSet (void) const [inline], [private]

Definition at line 140 of file Nullable.h.

Referenced by Rcpp::Nullable< T >::get(), Rcpp::Nullable< T >::isNull(), and Rcpp::Nullable< T >::operator SEXP().

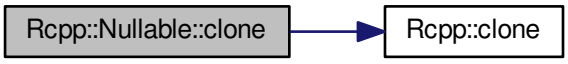
6.453.3.3 template<class T > T Rcpp::Nullable< T >::clone ( ) const [inline]

Return a clone of m_sexp as a T

Definition at line 134 of file Nullable.h.

References Rcpp::clone().

Here is the call graph for this function:



6.453.3.4 template<class T > SEXP Rcpp::Nullable< T >::get ( ) const [inline]

get() accessor for object

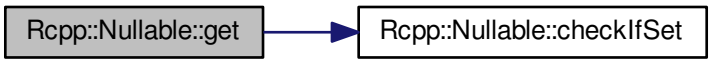
Exceptions

'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.

Here is the call graph for this function:



6.453.3.5 template<class T > bool Rcpp::Nullable< T >::isNotNull ( ) const [inline]

Boolean test for not NULL

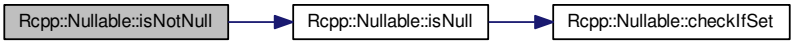
Exceptions

'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:



6.453.3.6 template<class T > bool Rcpp::Nullable< T >::isNull ( ) const [inline]

Boolean test for NULL

Exceptions

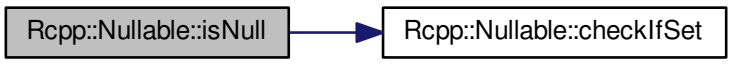
'not	initialized' if object has not been set

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:



6.453.3.7 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.

6.453.3.8 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.

6.453.3.9 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:

6.453.3.10 template<class T > Nullable& Rcpp::Nullable< T >::operator= (SEXP *sexp*) [inline]

Copy constructor for Nullable object

Parameters

SEXP	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.

6.453.4 Member Data Documentation

6.453.4.1 template<class T > bool Rcpp::Nullable< T >::m_set [private]

Definition at line 138 of file Nullable.h.

Referenced by Rcpp::Nullable< T >::isSet(), Rcpp::Nullable< T >::isUsable(), Rcpp::Nullable< T >::Nullable(), and Rcpp::Nullable< T >::operator=().

6.453.4.2 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

6.454 Rcpp::traits::num2type< N > Struct Template Reference

`#include <num2type.h>`

6.454.1 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

6.455 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

6.455.1 Detailed Description

template<typename T>
class Rcpp::object< T >

Definition at line 65 of file Module.h.

6.455.2 Member Typedef Documentation

6.455.2.1 template<typename T > typedef T Rcpp::object< T >::object_type

Definition at line 67 of file Module.h.

6.455.3 Constructor & Destructor Documentation

6.455.3.1 template<typename T > Rcpp::object< T >::object (T * *ptr_*) [inline]

Definition at line 68 of file Module.h.

6.455.4 Member Function Documentation

6.455.4.1 template<typename T > Rcpp::object< T >::operator T * ( ) [inline]

Definition at line 69 of file Module.h.

6.455.4.2 template<typename T > T& Rcpp::object< T >::operator& ( ) [inline]

Definition at line 71 of file Module.h.

6.455.4.3 template<typename T > T* Rcpp::object< T >::operator-> ( ) [inline]

Definition at line 70 of file Module.h.

6.455.5 Member Data Documentation

6.455.5.1 template<typename T > T* Rcpp::object< T >::ptr

Definition at line 72 of file Module.h.

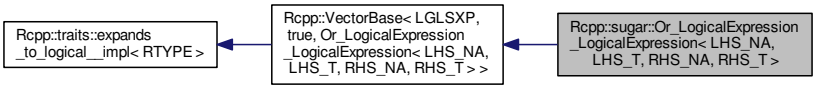
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

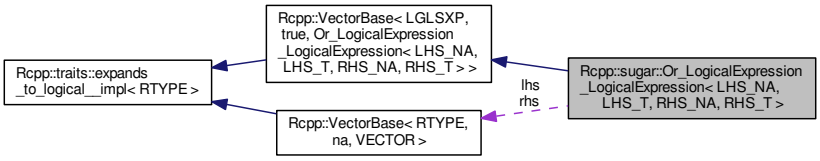
6.456 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.456.1 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.

6.456.2 Member Typedef Documentation

6.456.2.1 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.

6.456.2.2 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.

6.456.3 Constructor & Destructor Documentation

6.456.3.1 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.

6.456.4 Member Function Documentation

6.456.4.1 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.

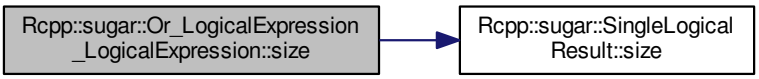
References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.456.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.456.5 Member Data Documentation

6.456.5.1 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.

6.456.5.2 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.

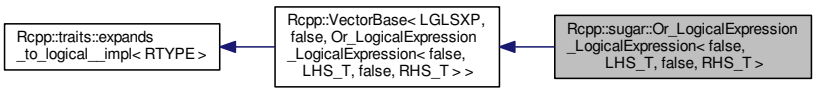
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

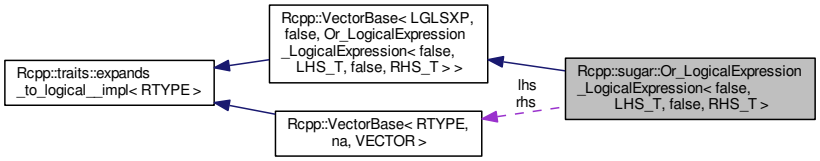
6.457 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.457.1 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.

6.457.2 Member Typedef Documentation

6.457.2.1 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.

6.457.2.2 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.

6.457.3 Constructor & Destructor Documentation

6.457.3.1 template<typename LHS_T , typename RHS_T > Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::Or_LogicalExpression_LogicalExpression (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 267 of file or.h.

6.457.4 Member Function Documentation

6.457.4.1 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.

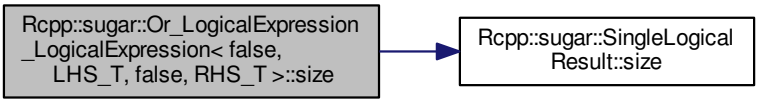
References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.457.4.2 template<typename LHS_T , typename RHS_T > R_xlen_t Rcpp::sugar::Or_LogicalExpression_Logical↩ Expression< false, LHS_T, false, RHS_T >::size ( ) const [inline]

Definition at line 273 of file or.h.

References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.457.5 Member Data Documentation

6.457.5.1 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.

6.457.5.2 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.

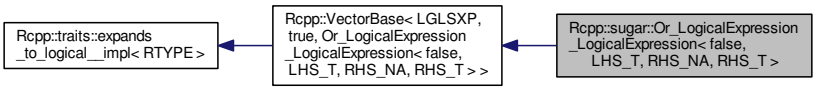
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

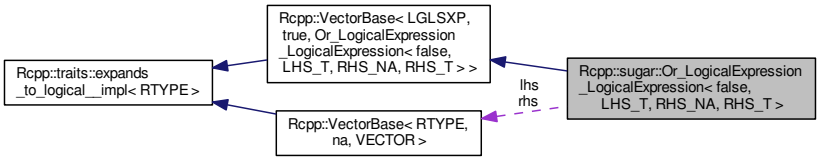
6.458 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.458.1 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.

6.458.2 Member Typedef Documentation

6.458.2.1 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.

6.458.2.2 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.

6.458.3 Constructor & Destructor Documentation

6.458.3.1 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.

6.458.4 Member Function Documentation

6.458.4.1 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.

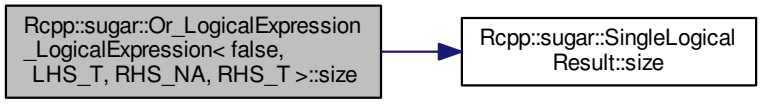
References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.458.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.458.5 Member Data Documentation

6.458.5.1 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.

6.458.5.2 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.

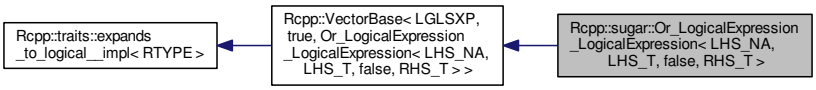
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

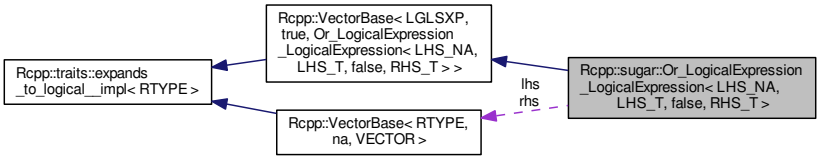
6.459 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.459.1 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.

6.459.2 Member Typedef Documentation

6.459.2.1 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.

6.459.2.2 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.

6.459.3 Constructor & Destructor Documentation

6.459.3.1 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.

6.459.4 Member Function Documentation

6.459.4.1 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.

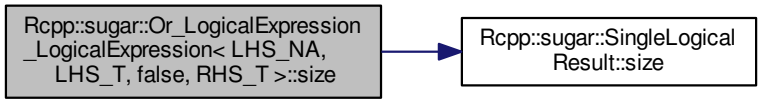
References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.459.4.2 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_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::SingleLogicalResult< NA, T >::size().

Here is the call graph for this function:



6.459.5 Member Data Documentation

6.459.5.1 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.

6.459.5.2 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.

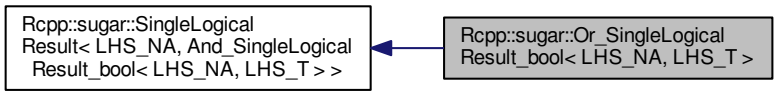
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

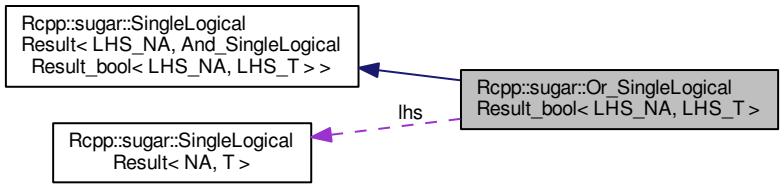
6.460 Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference

`#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

6.460.1 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.

6.460.2 Member Typedef Documentation

6.460.2.1 template<bool LHS_NA, typename LHS_T> typedef SingleLogicalResult< LHS_NA , Or_SingleLogical↩ Result_bool<LHS_NA,LHS_T> > Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::BASE

Definition at line 181 of file or.h.

6.460.2.2 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.

6.460.3 Constructor & Destructor Documentation

6.460.3.1 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.

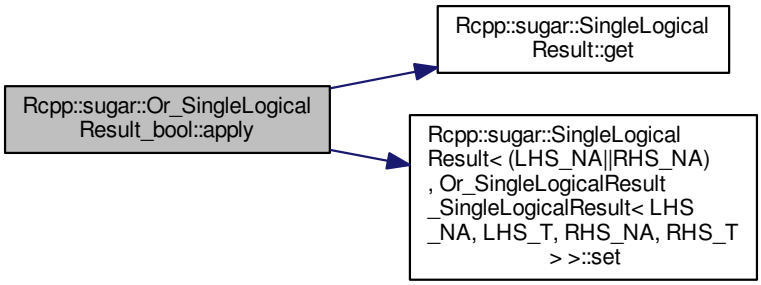
6.460.4 Member Function Documentation

6.460.4.1 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_SingleLogical↩ Result< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< L↩ HS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult< (LHS_NA||RHS_NA), Or_Single↩ LogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:



6.460.5 Member Data Documentation

6.460.5.1 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.

6.460.5.2 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.

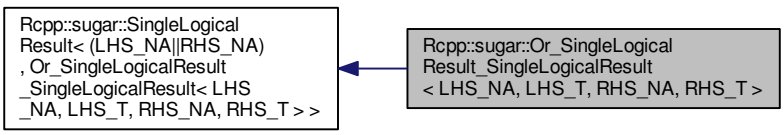
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

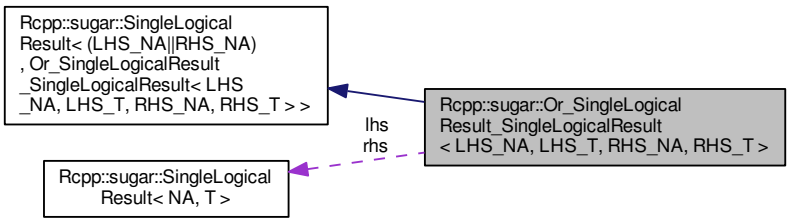
6.461 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.461.1 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.

6.461.2 Member Typedef Documentation

6.461.2.1 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.

6.461.2.2 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.

6.461.2.3 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.

6.461.3 Constructor & Destructor Documentation

6.461.3.1 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_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 43 of file or.h.

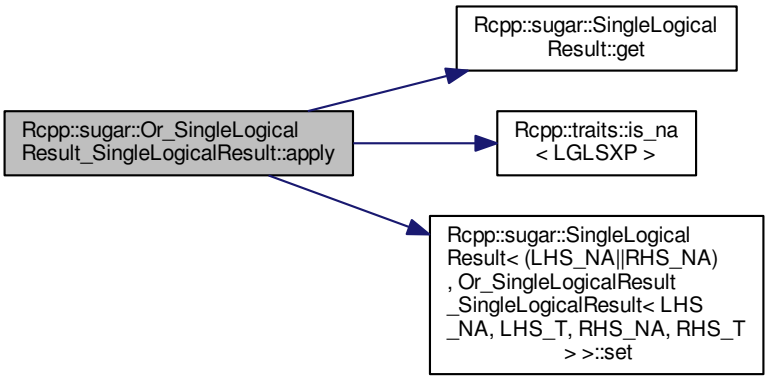
6.461.4 Member Function Documentation

6.461.4.1 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:



6.461.5 Member Data Documentation

6.461.5.1 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(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), 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[](), operator|(), 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().

6.461.5.2 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(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), 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[](), operator|(), and operator||().

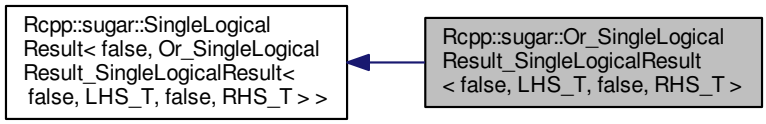
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

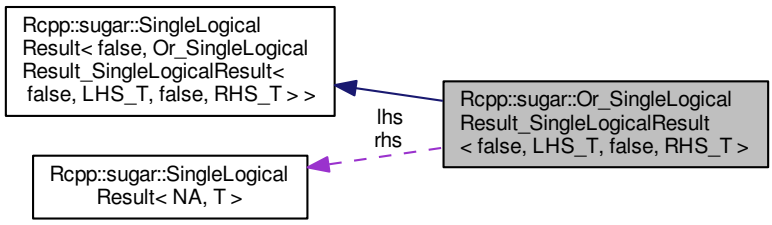
6.462 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.462.1 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.

6.462.2 Member Typedef Documentation

6.462.2.1 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.

6.462.2.2 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.

6.462.2.3 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.

6.462.3 Constructor & Destructor Documentation

6.462.3.1 template<typename LHS_T , typename RHS_T > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 150 of file or.h.

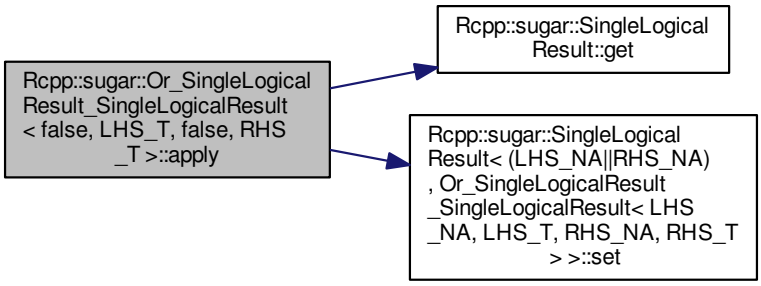
6.462.4 Member Function Documentation

6.462.4.1 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:



6.462.5 Member Data Documentation

6.462.5.1 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.

6.462.5.2 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.

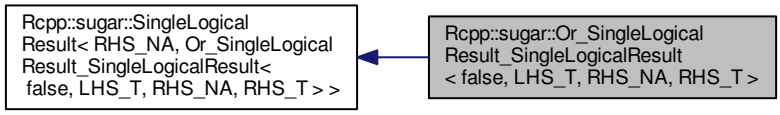
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

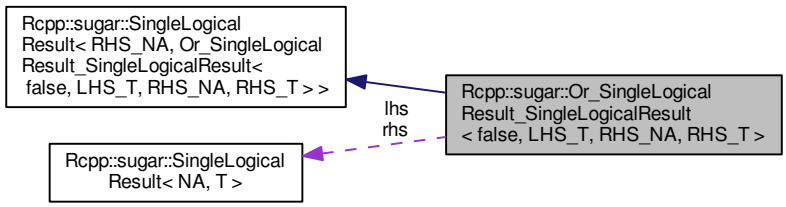
6.463 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.463.1 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.

6.463.2 Member Typedef Documentation

6.463.2.1 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.

6.463.2.2 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.

6.463.2.3 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.

6.463.3 Constructor & Destructor Documentation

6.463.3.1 template<typename LHS_T , bool RHS_NA, typename RHS_T > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 115 of file or.h.

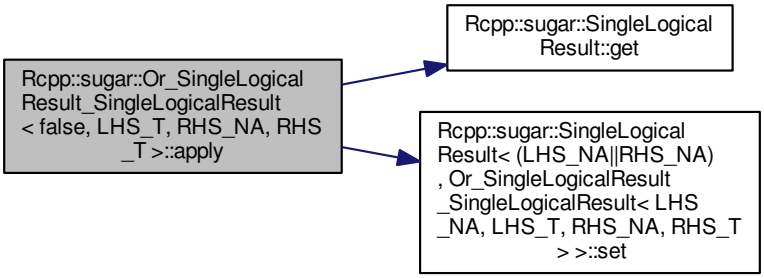
6.463.4 Member Function Documentation

6.463.4.1 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:



6.463.5 Member Data Documentation

6.463.5.1 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.

6.463.5.2 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.

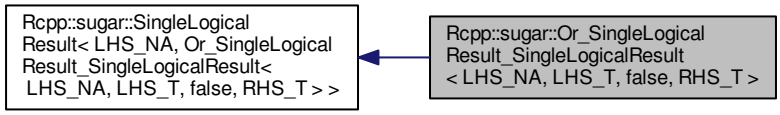
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

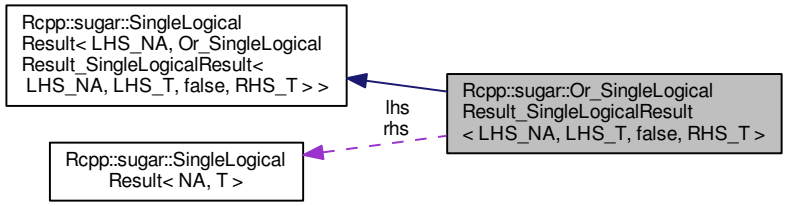
6.464 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.464.1 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.

6.464.2 Member Typedef Documentation

6.464.2.1 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.

6.464.2.2 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.

6.464.2.3 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.

6.464.3 Constructor & Destructor Documentation

6.464.3.1 template<bool LHS_NA, typename LHS_T , typename RHS_T > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 79 of file or.h.

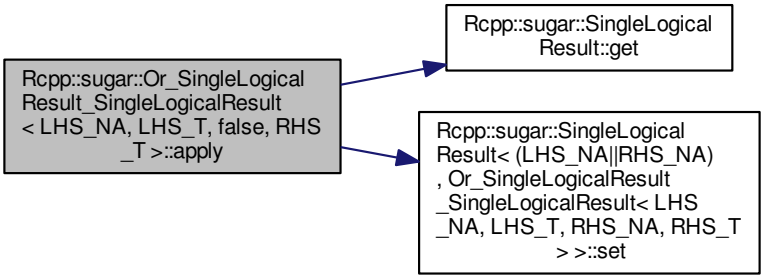
6.464.4 Member Function Documentation

6.464.4.1 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_SingleLogical↩ Result< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< L↩ HS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult< (LHS_NA||RHS_NA), Or_Single↩ LogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:



6.464.5 Member Data Documentation

6.464.5.1 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.

6.464.5.2 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.

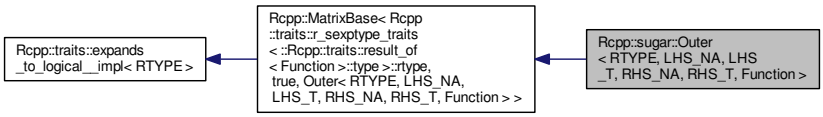
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

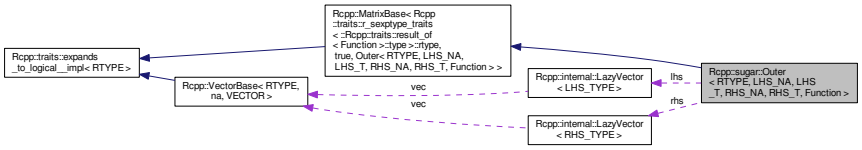
6.465 Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > Class Template Reference

`#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

6.465.1 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.

6.465.2 Member Typedef Documentation

6.465.2.1 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.

6.465.2.2 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.

6.465.2.3 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.

6.465.2.4 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.

6.465.2.5 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.

6.465.2.6 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.

6.465.2.7 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.

6.465.3 Constructor & Destructor Documentation

6.465.3.1 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.

6.465.4 Member Function Documentation

6.465.4.1 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.

6.465.4.2 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.

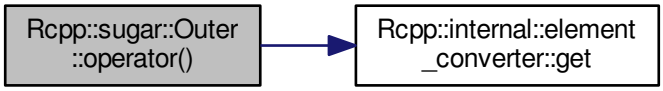
References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nr.

6.465.4.3 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:



6.465.4.4 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.

6.465.5 Member Data Documentation

6.465.5.1 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()(), and Rcpp::outer().

6.465.5.2 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()(), and Rcpp::outer().

6.465.5.3 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().

6.465.5.4 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().

6.465.5.5 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.

6.465.5.6 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()(), and Rcpp↩ ::outer().

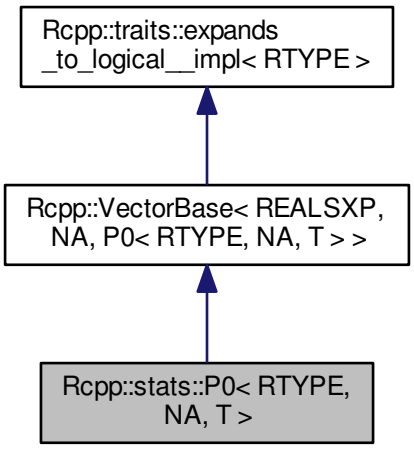
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/outer.h

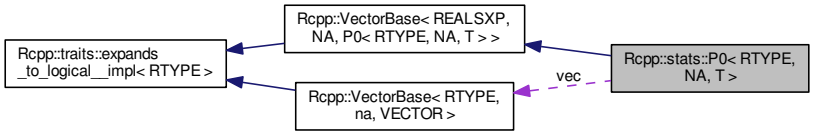
6.466 Rcpp::stats::P0< RTYPE, NA, T > Class Template Reference

`#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

6.466.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P0< RTYPE, NA, T >

Definition at line 123 of file dpq.h.

6.466.2 Member Typedef Documentation

6.466.2.1 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.

6.466.2.2 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.

6.466.3 Constructor & Destructor Documentation

6.466.3.1 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.

6.466.4 Member Function Documentation

6.466.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.466.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.466.5 Member Data Documentation

6.466.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P0< RTYPE, NA, T >::log [private]

Definition at line 141 of file dpq.h.

6.466.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P0< RTYPE, NA, T >::lower [private]

Definition at line 141 of file dpq.h.

6.466.5.3 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::P0< RTYPE, NA, T >::ptr [private]

Definition at line 139 of file dpq.h.

6.466.5.4 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.

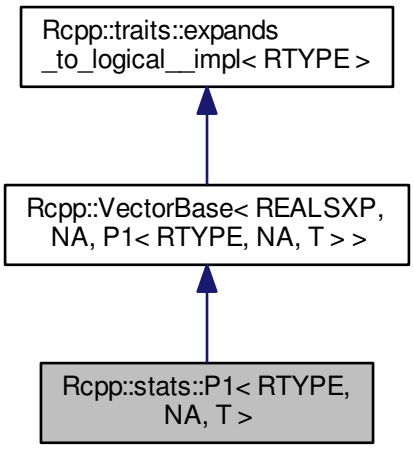
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

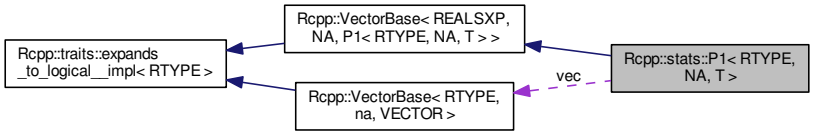
6.467 Rcpp::stats::P1< RTYPE, NA, T > Class Template Reference

`#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

6.467.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P1< RTYPE, NA, T >

Definition at line 147 of file dpq.h.

6.467.2 Member Typedef Documentation

6.467.2.1 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.

6.467.2.2 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.

6.467.3 Constructor & Destructor Documentation

6.467.3.1 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.

6.467.4 Member Function Documentation

6.467.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.467.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.467.5 Member Data Documentation

6.467.5.1 template<int RTYPE, bool NA, typename T> int Rcpp::stats::P1< RTYPE, NA, T >::log [private]

Definition at line 166 of file dpq.h.

6.467.5.2 template<int RTYPE, bool NA, typename T> int Rcpp::stats::P1< RTYPE, NA, T >::lower [private]

Definition at line 166 of file dpq.h.

6.467.5.3 template<int RTYPE, bool NA, typename T> double Rcpp::stats::P1< RTYPE, NA, T >::p0 [private]

Definition at line 165 of file dpq.h.

6.467.5.4 template<int RTYPE, bool NA, typename T> FunPtr Rcpp::stats::P1< RTYPE, NA, T >::ptr [private]

Definition at line 163 of file dpq.h.

6.467.5.5 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.

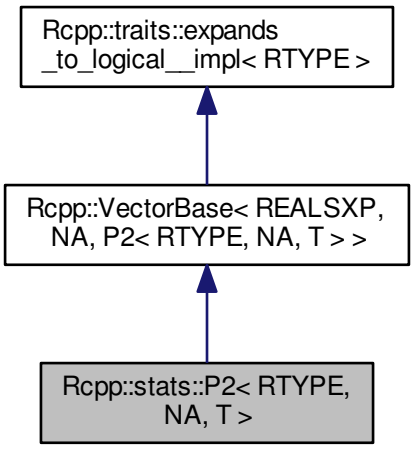
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

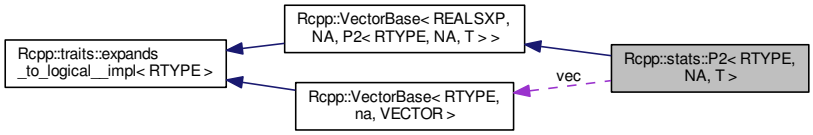
6.468 Rcpp::stats::P2< RTYPE, NA, T > Class Template Reference

`#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

6.468.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P2< RTYPE, NA, T >

Definition at line 172 of file dpq.h.

6.468.2 Member Typedef Documentation

6.468.2.1 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.

6.468.2.2 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.

6.468.3 Constructor & Destructor Documentation

6.468.3.1 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.

6.468.4 Member Function Documentation

6.468.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.468.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.468.5 Member Data Documentation

6.468.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P2< RTYPE, NA, T >::log [private]

Definition at line 191 of file dpq.h.

6.468.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P2< RTYPE, NA, T >::lower [private]

Definition at line 191 of file dpq.h.

6.468.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::P2< RTYPE, NA, T >::p0 [private]

Definition at line 190 of file dpq.h.

6.468.5.4 template<int RTYPE, bool NA, typename T > double Rcpp::stats::P2< RTYPE, NA, T >::p1 [private]

Definition at line 190 of file dpq.h.

6.468.5.5 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::P2< RTYPE, NA, T >::ptr [private]

Definition at line 188 of file dpq.h.

6.468.5.6 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.

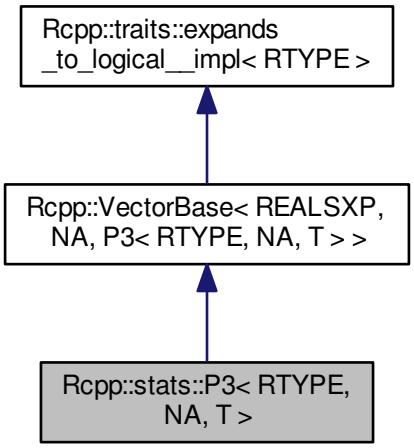
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

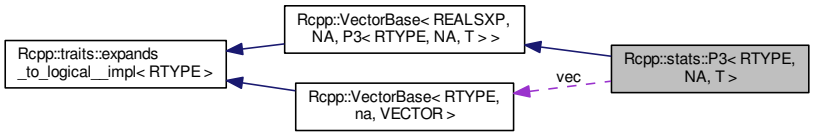
6.469 Rcpp::stats::P3< RTYPE, NA, T > Class Template Reference

`#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

6.469.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P3< RTYPE, NA, T >

Definition at line 195 of file dpq.h.

6.469.2 Member Typedef Documentation

6.469.2.1 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.

6.469.2.2 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.

6.469.3 Constructor & Destructor Documentation

6.469.3.1 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.

6.469.4 Member Function Documentation

6.469.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.469.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.469.5 Member Data Documentation

6.469.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P3< RTYPE, NA, T >::log [private]

Definition at line 214 of file dpq.h.

6.469.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::P3< RTYPE, NA, T >::lower [private]

Definition at line 214 of file dpq.h.

6.469.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::P3< RTYPE, NA, T >::p0 [private]

Definition at line 213 of file dpq.h.

6.469.5.4 template<int RTYPE, bool NA, typename T > double Rcpp::stats::P3< RTYPE, NA, T >::p1 [private]

Definition at line 213 of file dpq.h.

6.469.5.5 template<int RTYPE, bool NA, typename T > double Rcpp::stats::P3< RTYPE, NA, T >::p2 [private]

Definition at line 213 of file dpq.h.

6.469.5.6 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::P3< RTYPE, NA, T >::ptr [private]

Definition at line 211 of file dpq.h.

6.469.5.7 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

6.470 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_

6.470.1 Detailed Description

Definition at line 281 of file attributes.cpp.

6.470.2 Constructor & Destructor Documentation

6.470.2.1 Rcpp::attributes::Param::Param ( ) [inline]

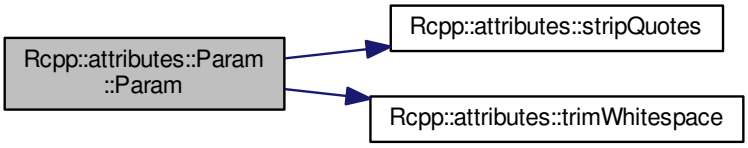
Definition at line 283 of file attributes.cpp.

6.470.2.2 Rcpp::attributes::Param::Param (const std::string & *paramText*) [explicit]

Definition at line 993 of file attributes.cpp.

References Rcpp::attributes::stripQuotes(), and Rcpp::attributes::trimWhitespace().

Here is the call graph for this function:



6.470.3 Member Function Documentation

6.470.3.1 bool Rcpp::attributes::Param::empty ( ) const [inline]

Definition at line 285 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::hasParameter(), Rcpp::attributes::operator<<(), and Rcpp::attributes::↩ Attribute::rng().

6.470.3.2 const std::string& Rcpp::attributes::Param::name ( ) const [inline]

Definition at line 297 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.470.3.3 bool Rcpp::attributes::Param::operator!= (const Param & *other*) const [inline]

Definition at line 292 of file attributes.cpp.

6.470.3.4 bool Rcpp::attributes::Param::operator== (const Param & *other*) const [inline]

Definition at line 287 of file attributes.cpp.

References name_, and value_.

6.470.3.5 const std::string& Rcpp::attributes::Param::value ( ) const [inline]

Definition at line 298 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::exportedName(), Rcpp::attributes::operator<<(), and Rcpp::attributes::Attribute::rng().

6.470.4 Member Data Documentation

6.470.4.1 std::string Rcpp::attributes::Param::name_ [private]

Definition at line 301 of file attributes.cpp.

Referenced by operator==().

6.470.4.2 std::string Rcpp::attributes::Param::value_ [private]

Definition at line 302 of file attributes.cpp.

Referenced by operator==().

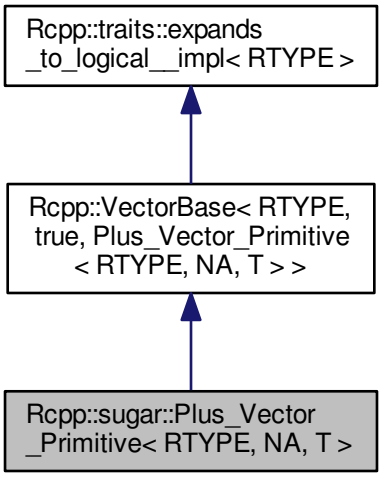
The documentation for this class was generated from the following file:

- src/attributes.cpp

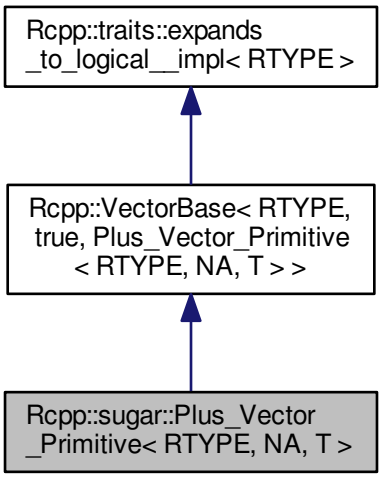
6.471 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T > Class Template Reference

`#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

6.471.1 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.

6.471.2 Member Typedef Documentation

6.471.2.1 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.

6.471.2.2 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.

6.471.2.3 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.

6.471.3 Constructor & Destructor Documentation

6.471.3.1 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.

6.471.4 Member Function Documentation

6.471.4.1 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_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.

6.471.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.471.5 Member Data Documentation

6.471.5.1 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.

6.471.5.2 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.

6.471.5.3 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.

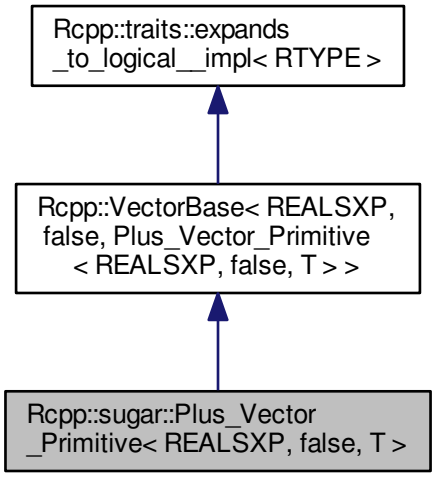
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

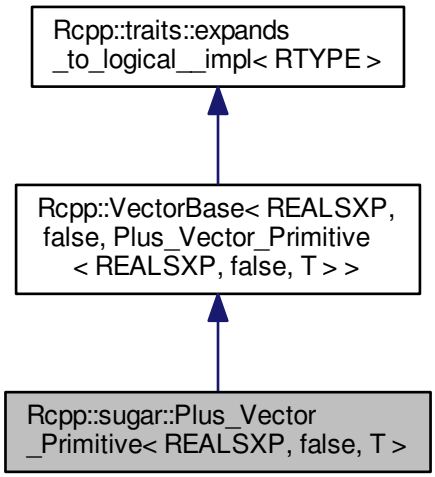
6.472 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T > Class Template Reference

`#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

6.472.1 Detailed Description

template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >

Definition at line 323 of file plus.h.

6.472.2 Member Typedef Documentation

6.472.2.1 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.

6.472.2.2 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.

6.472.3 Constructor & Destructor Documentation

6.472.3.1 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.

6.472.4 Member Function Documentation

6.472.4.1 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_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.

6.472.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.472.5 Member Data Documentation

6.472.5.1 template<typename T > const EXT& Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::lhs [private]

Definition at line 340 of file plus.h.

6.472.5.2 template<typename T > double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::rhs [private]

Definition at line 341 of file plus.h.

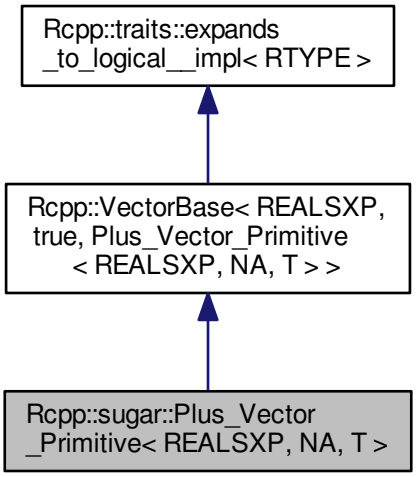
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

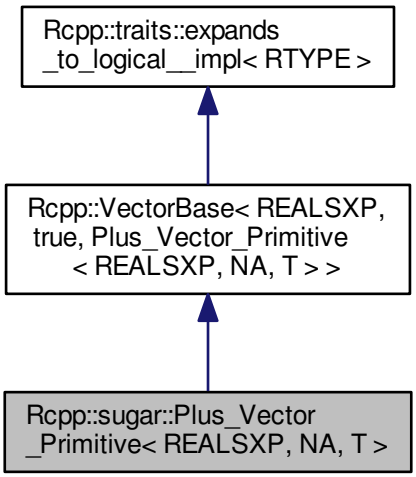
6.473 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T > Class Template Reference

`#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

6.473.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >

Definition at line 276 of file plus.h.

6.473.2 Member Typedef Documentation

6.473.2.1 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.

6.473.2.2 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.

6.473.3 Constructor & Destructor Documentation

6.473.3.1 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.

6.473.4 Member Function Documentation

6.473.4.1 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_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.

6.473.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.473.5 Member Data Documentation

6.473.5.1 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.

6.473.5.2 template<bool NA, typename T > double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::rhs [private]

Definition at line 294 of file plus.h.

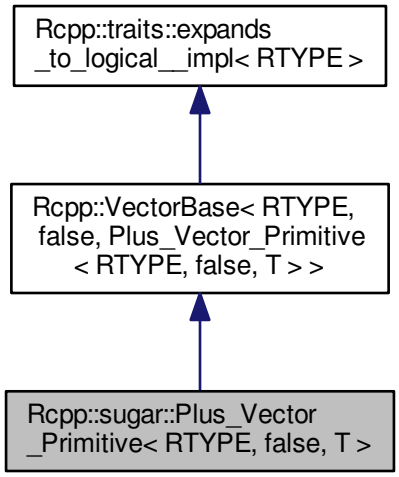
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

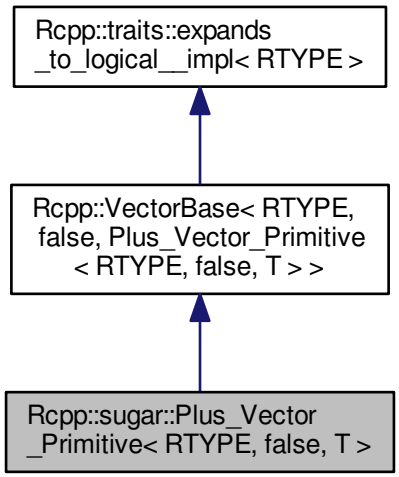
6.474 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T > Class Template Reference

`#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

6.474.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >

Definition at line 300 of file plus.h.

6.474.2 Member Typedef Documentation

6.474.2.1 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.

6.474.2.2 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.

6.474.2.3 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.

6.474.3 Constructor & Destructor Documentation

6.474.3.1 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.

6.474.4 Member Function Documentation

6.474.4.1 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_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.

6.474.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.474.5 Member Data Documentation

6.474.5.1 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.

6.474.5.2 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::rhs [private]

Definition at line 318 of file plus.h.

6.474.5.3 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.

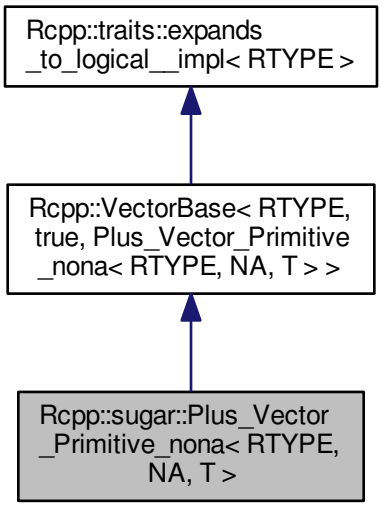
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

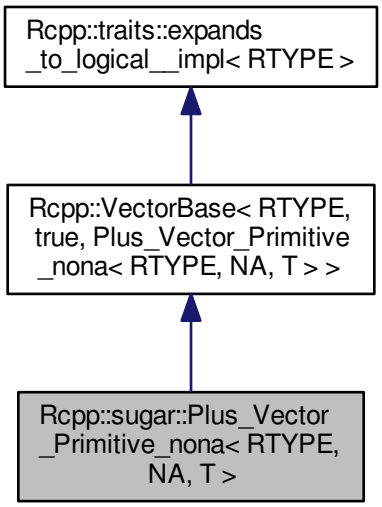
6.475 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference

`#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

6.475.1 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.

6.475.2 Member Typedef Documentation

6.475.2.1 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.

6.475.2.2 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.

6.475.2.3 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.

6.475.3 Constructor & Destructor Documentation

6.475.3.1 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.

6.475.4 Member Function Documentation

6.475.4.1 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_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.

6.475.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.475.5 Member Data Documentation

6.475.5.1 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.

6.475.5.2 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.

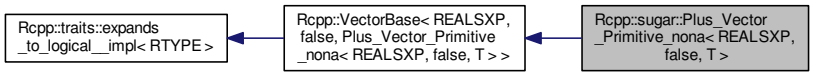
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

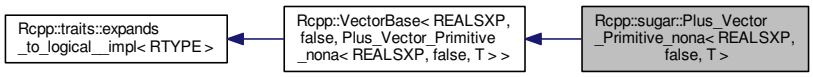
6.476 Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T > Class Template Reference

`#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

6.476.1 Detailed Description

template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >

Definition at line 422 of file plus.h.

6.476.2 Member Typedef Documentation

6.476.2.1 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.

6.476.2.2 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.

6.476.3 Constructor & Destructor Documentation

6.476.3.1 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.

6.476.4 Member Function Documentation

6.476.4.1 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_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.

6.476.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.476.5 Member Data Documentation

6.476.5.1 template<typename T > const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::lhs [private]

Definition at line 438 of file plus.h.

6.476.5.2 template<typename T > double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::rhs [private]

Definition at line 439 of file plus.h.

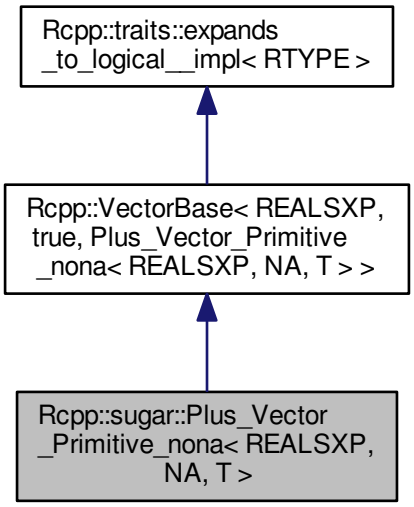
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

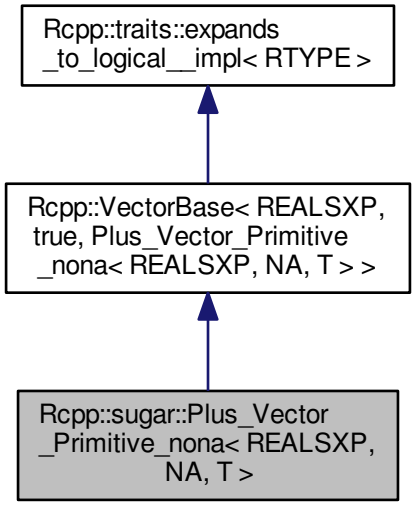
6.477 Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T > Class Template Reference

`#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

6.477.1 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.

6.477.2 Member Typedef Documentation

6.477.2.1 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.

6.477.2.2 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.

6.477.3 Constructor & Destructor Documentation

6.477.3.1 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.

6.477.4 Member Function Documentation

6.477.4.1 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_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.

6.477.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.477.5 Member Data Documentation

6.477.5.1 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.

6.477.5.2 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.

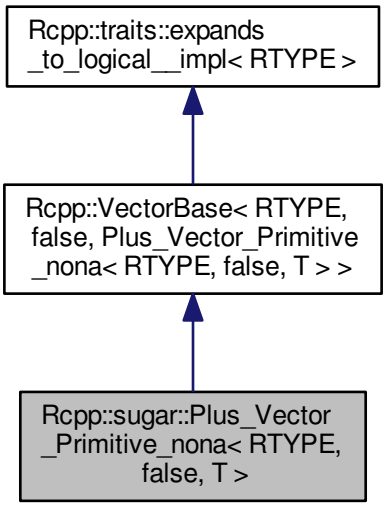
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

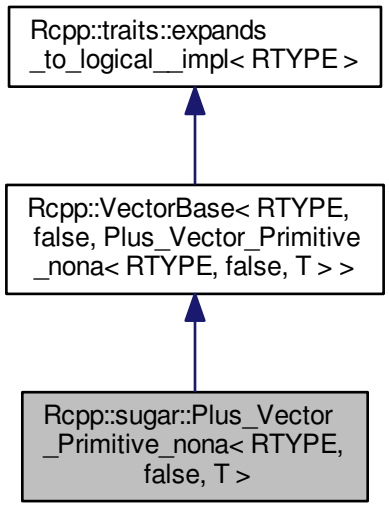
6.478 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference

`#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

6.478.1 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.

6.478.2 Member Typedef Documentation

6.478.2.1 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.

6.478.2.2 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.

6.478.2.3 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.

6.478.3 Constructor & Destructor Documentation

6.478.3.1 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.

6.478.4 Member Function Documentation

6.478.4.1 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_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.

6.478.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.478.5 Member Data Documentation

6.478.5.1 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.

6.478.5.2 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.

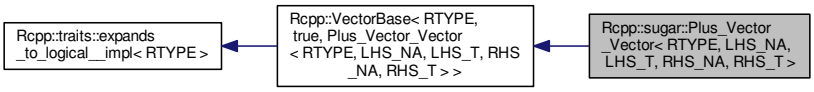
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

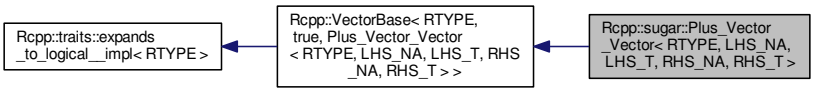
6.479 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.479.1 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.

6.479.2 Member Typedef Documentation

6.479.2.1 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.

6.479.2.2 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.

6.479.2.3 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.

6.479.2.4 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.

6.479.2.5 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.

6.479.3 Constructor & Destructor Documentation

6.479.3.1 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.

6.479.4 Member Function Documentation

6.479.4.1 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.

6.479.4.2 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.

6.479.5 Member Data Documentation

6.479.5.1 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::operator+(), 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_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_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_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(), Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size(), 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(), Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::size(), 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().

6.479.5.2 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::operator+(), 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_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_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[]().

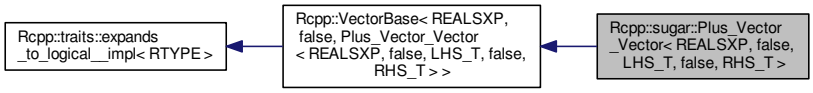
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

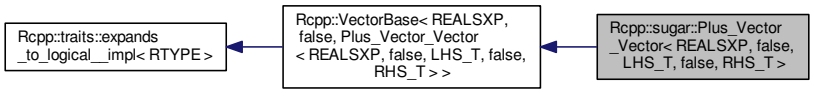
6.480 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.480.1 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.

6.480.2 Member Typedef Documentation

6.480.2.1 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.

6.480.2.2 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.

6.480.2.3 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.

6.480.2.4 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.

6.480.3 Constructor & Destructor Documentation

6.480.3.1 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.

6.480.4 Member Function Documentation

6.480.4.1 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.

6.480.4.2 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.

6.480.5 Member Data Documentation

6.480.5.1 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.

6.480.5.2 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.

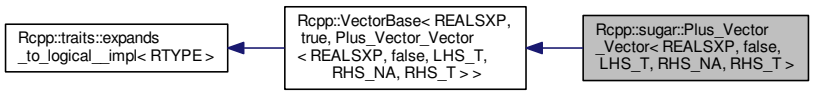
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

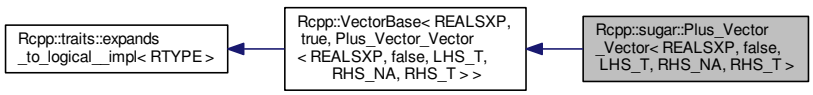
6.481 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.481.1 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.

6.481.2 Member Typedef Documentation

6.481.2.1 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.

6.481.2.2 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.

6.481.2.3 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.

6.481.2.4 template<typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<REAL↩ SXP,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.

6.481.3 Constructor & Destructor Documentation

6.481.3.1 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.

6.481.4 Member Function Documentation

6.481.4.1 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.

6.481.4.2 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.

6.481.5 Member Data Documentation

6.481.5.1 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.

6.481.5.2 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.

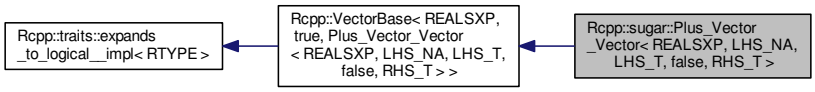
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

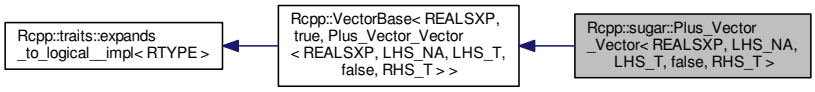
6.482 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.482.1 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.

6.482.2 Member Typedef Documentation

6.482.2.1 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.

6.482.2.2 template<bool LHS_NA, typename LHS_T , typename RHS_T > typedef Rcpp::VectorBase<REALS↩ XP,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.

6.482.2.3 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.

6.482.2.4 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.

6.482.3 Constructor & Destructor Documentation

6.482.3.1 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 LHS_TYPE & *lhs_*, const RHS_TYPE & *rhs_*) [inline]

Definition at line 175 of file plus.h.

6.482.4 Member Function Documentation

6.482.4.1 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.

6.482.4.2 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.

6.482.5 Member Data Documentation

6.482.5.1 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.

6.482.5.2 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.

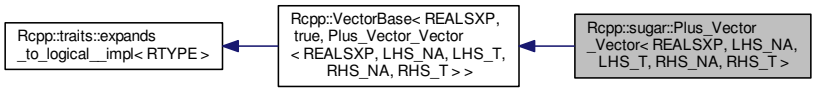
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

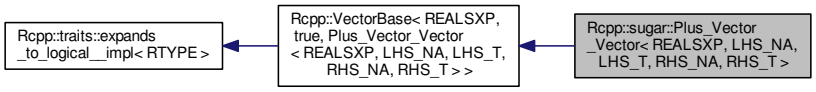
6.483 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.483.1 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.

6.483.2 Member Typedef Documentation

6.483.2.1 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.

6.483.2.2 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.

6.483.2.3 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.

6.483.2.4 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.

6.483.3 Constructor & Destructor Documentation

6.483.3.1 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.

6.483.4 Member Function Documentation

6.483.4.1 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.

6.483.4.2 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.

6.483.5 Member Data Documentation

6.483.5.1 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.

6.483.5.2 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.

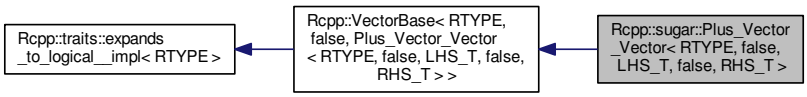
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

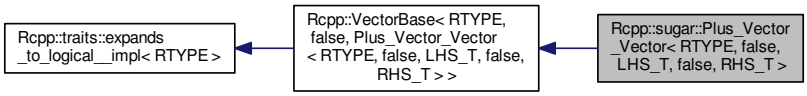
6.484 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.484.1 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.

6.484.2 Member Typedef Documentation

6.484.2.1 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.

6.484.2.2 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.

6.484.2.3 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.

6.484.2.4 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.

6.484.2.5 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.

6.484.3 Constructor & Destructor Documentation

6.484.3.1 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.

6.484.4 Member Function Documentation

6.484.4.1 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.

6.484.4.2 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.

6.484.5 Member Data Documentation

6.484.5.1 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.

6.484.5.2 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.

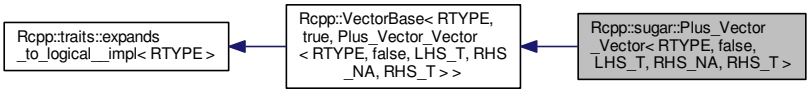
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

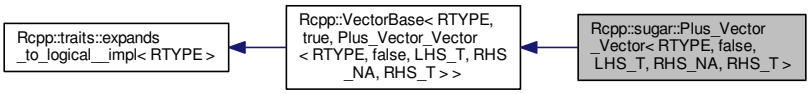
6.485 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.485.1 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.

6.485.2 Member Typedef Documentation

6.485.2.1 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.

6.485.2.2 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::Vector↩ Base<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.

6.485.2.3 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.

6.485.2.4 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.485.2.5 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.

6.485.3 Constructor & Destructor Documentation

6.485.3.1 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.

6.485.4 Member Function Documentation

6.485.4.1 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.

6.485.4.2 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.

6.485.5 Member Data Documentation

6.485.5.1 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.

6.485.5.2 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.

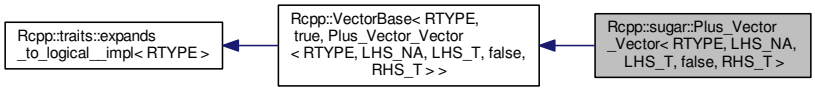
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

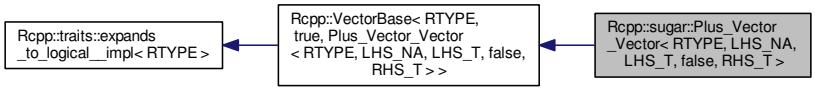
6.486 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.486.1 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.

6.486.2 Member Typedef Documentation

6.486.2.1 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.

6.486.2.2 template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.486.2.3 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.

6.486.2.4 template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T > typedef Rcpp::Vector↩ Base<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.

6.486.2.5 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.

6.486.3 Constructor & Destructor Documentation

6.486.3.1 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.

6.486.4 Member Function Documentation

6.486.4.1 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.

6.486.4.2 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.

6.486.5 Member Data Documentation

6.486.5.1 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.

6.486.5.2 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

6.487 Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference

```
#include <pmax.h>
```

6.487.1 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

6.488 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

6.488.1 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.

6.488.2 Member Function Documentation

6.488.2.1 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

6.489 Rcpp::sugar::pmax_op< REALSXP, false, false > Struct Template Reference

`#include <pmax.h>`

Public Member Functions

- double operator() (double left, double right) const

6.489.1 Detailed Description

template<>
struct Rcpp::sugar::pmax_op< REALSXP, false, false >

Definition at line 48 of file pmax.h.

6.489.2 Member Function Documentation

6.489.2.1 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

6.490 Rcpp::sugar::pmax_op< REALSXP, false, true > Struct Template Reference

`#include <pmax.h>`

Public Member Functions

- double operator() (double left, double right) const

6.490.1 Detailed Description

template<>
struct Rcpp::sugar::pmax_op< REALSXP, false, true >

Definition at line 43 of file pmax.h.

6.490.2 Member Function Documentation

6.490.2.1 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

6.491 Rcpp::sugar::pmax_op< REALSXP, true, false > Struct Template Reference

`#include <pmax.h>`

Public Member Functions

- double operator() (double left, double right) const

6.491.1 Detailed Description

template<>
struct Rcpp::sugar::pmax_op< REALSXP, true, false >

Definition at line 38 of file pmax.h.

6.491.2 Member Function Documentation

6.491.2.1 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

6.492 Rcpp::sugar::pmax_op< REALSXP, true, true > Struct Template Reference

`#include <pmax.h>`

Public Member Functions

- double operator() (double left, double right) const

6.492.1 Detailed Description

template<>
struct Rcpp::sugar::pmax_op< REALSXP, true, true >

Definition at line 33 of file pmax.h.

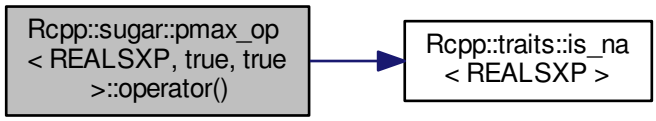
6.492.2 Member Function Documentation

6.492.2.1 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

6.493 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

6.493.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >

Definition at line 65 of file pmax.h.

6.493.2 Member Typedef Documentation

6.493.2.1 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.

6.493.3 Constructor & Destructor Documentation

6.493.3.1 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.

6.493.4 Member Function Documentation

6.493.4.1 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.

6.493.5 Member Data Documentation

6.493.5.1 template<int RTYPE, bool NA> STORAGE Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::right [private]

Definition at line 76 of file pmax.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.494 Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true > Class Template Reference

`#include <pmax.h>`

Public Member Functions

- pmax_op_Vector_Primitive (double right_)
- double operator() (double left) const

Private Attributes

- double right

6.494.1 Detailed Description

template<>
class Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >

Definition at line 79 of file pmax.h.

6.494.2 Constructor & Destructor Documentation

6.494.2.1 Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::pmax_op_Vector_Primitive (double *right_*) [inline]

Definition at line 81 of file pmax.h.

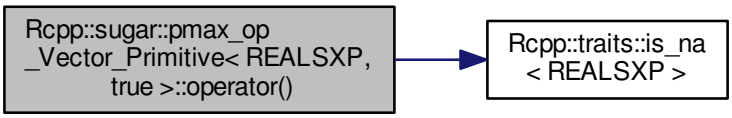
6.494.3 Member Function Documentation

6.494.3.1 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 >().

Here is the call graph for this function:



6.494.4 Member Data Documentation

6.494.4.1 double Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::right [private]

Definition at line 88 of file pmax.h.

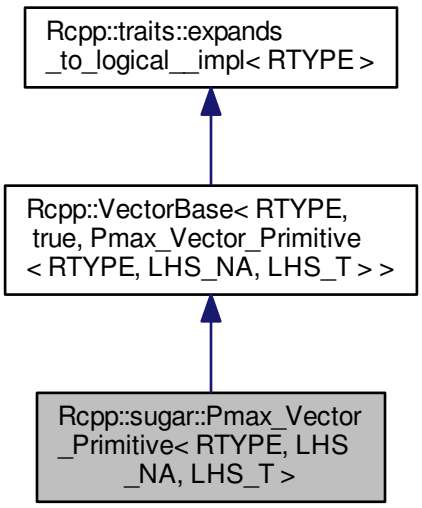
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

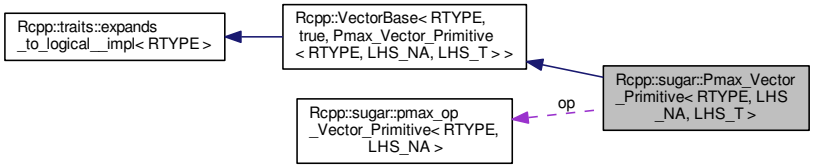
6.495 Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#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

6.495.1 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.

6.495.2 Member Typedef Documentation

6.495.2.1 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.

6.495.2.2 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.

6.495.3 Constructor & Destructor Documentation

6.495.3.1 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.

6.495.4 Member Function Documentation

6.495.4.1 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.

6.495.4.2 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.

6.495.5 Member Data Documentation

6.495.5.1 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.

6.495.5.2 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.

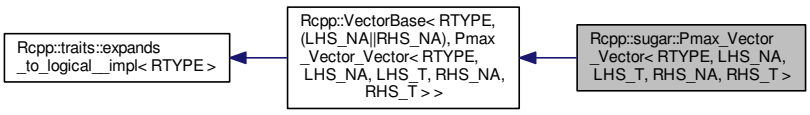
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

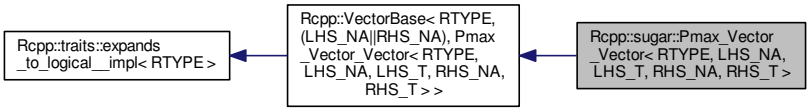
6.496 Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.496.1 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.

6.496.2 Member Typedef Documentation

6.496.2.1 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.

6.496.2.2 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.

6.496.3 Constructor & Destructor Documentation

6.496.3.1 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.

6.496.4 Member Function Documentation

6.496.4.1 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.

6.496.4.2 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.

6.496.5 Member Data Documentation

6.496.5.1 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.

6.496.5.2 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.

6.496.5.3 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.497 Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference

```
#include <pmin.h>
```

6.497.1 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

6.498 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

6.498.1 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.

6.498.2 Member Function Documentation

6.498.2.1 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

6.499 Rcpp::sugar::pmin_op< REALSXP, false, false > Struct Template Reference

`#include <pmin.h>`

Public Member Functions

- double operator() (double left, double right) const

6.499.1 Detailed Description

template<>
struct Rcpp::sugar::pmin_op< REALSXP, false, false >

Definition at line 48 of file pmin.h.

6.499.2 Member Function Documentation

6.499.2.1 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

6.500 Rcpp::sugar::pmin_op< REALSXP, false, true > Struct Template Reference

`#include <pmin.h>`

Public Member Functions

- double operator() (double left, double right) const

6.500.1 Detailed Description

template<>
struct Rcpp::sugar::pmin_op< REALSXP, false, true >

Definition at line 43 of file pmin.h.

6.500.2 Member Function Documentation

6.500.2.1 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

6.501 Rcpp::sugar::pmin_op< REALSXP, true, false > Struct Template Reference

`#include <pmin.h>`

Public Member Functions

- double operator() (double left, double right) const

6.501.1 Detailed Description

template<>
struct Rcpp::sugar::pmin_op< REALSXP, true, false >

Definition at line 38 of file pmin.h.

6.501.2 Member Function Documentation

6.501.2.1 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

6.502 Rcpp::sugar::pmin_op< REALSXP, true, true > Struct Template Reference

`#include <pmin.h>`

Public Member Functions

- double operator() (double left, double right) const

6.502.1 Detailed Description

template<>
struct Rcpp::sugar::pmin_op< REALSXP, true, true >

Definition at line 33 of file pmin.h.

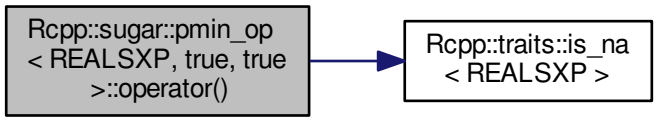
6.502.2 Member Function Documentation

6.502.2.1 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

6.503 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

6.503.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >

Definition at line 65 of file pmin.h.

6.503.2 Member Typedef Documentation

6.503.2.1 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.

6.503.3 Constructor & Destructor Documentation

6.503.3.1 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.

6.503.4 Member Function Documentation

6.503.4.1 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.

6.503.5 Member Data Documentation

6.503.5.1 template<int RTYPE, bool NA> STORAGE Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::right [private]

Definition at line 76 of file pmin.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.504 Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true > Class Template Reference

`#include <pmin.h>`

Public Member Functions

- pmin_op_Vector_Primitive (double right_)
- double operator() (double left) const

Private Attributes

- double right

6.504.1 Detailed Description

template<>
class Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >

Definition at line 79 of file pmin.h.

6.504.2 Constructor & Destructor Documentation

6.504.2.1 Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::pmin_op_Vector_Primitive (double *right_*) [inline]

Definition at line 81 of file pmin.h.

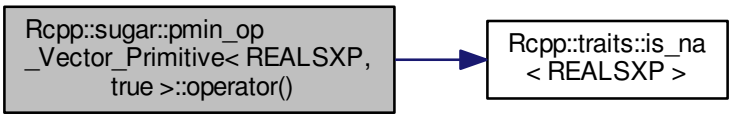
6.504.3 Member Function Documentation

6.504.3.1 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 >().

Here is the call graph for this function:



6.504.4 Member Data Documentation

6.504.4.1 double Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::right [private]

Definition at line 88 of file pmin.h.

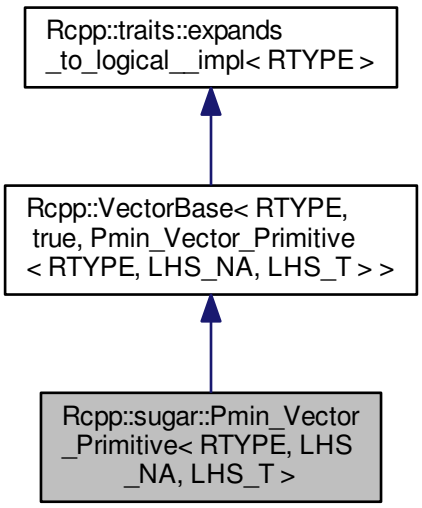
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

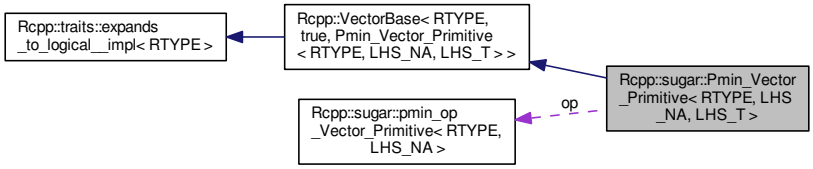
6.505 Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#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

6.505.1 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.

6.505.2 Member Typedef Documentation

6.505.2.1 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.

6.505.2.2 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.

6.505.3 Constructor & Destructor Documentation

6.505.3.1 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.

6.505.4 Member Function Documentation

6.505.4.1 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.

6.505.4.2 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.

6.505.5 Member Data Documentation

6.505.5.1 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.

6.505.5.2 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.

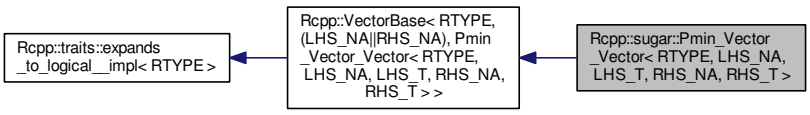
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

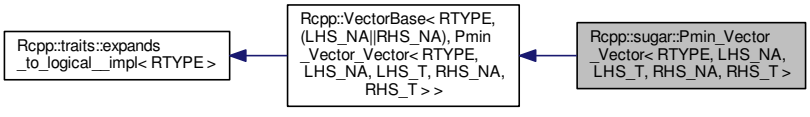
6.506 Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.506.1 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.

6.506.2 Member Typedef Documentation

6.506.2.1 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.

6.506.2.2 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.

6.506.3 Constructor & Destructor Documentation

6.506.3.1 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.

6.506.4 Member Function Documentation

6.506.4.1 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.

6.506.4.2 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.

6.506.5 Member Data Documentation

6.506.5.1 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.

6.506.5.2 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.

6.506.5.3 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.507 Rcpp::traits::pointer_wrap_tag Struct Reference

```
#include <module_wrap_traits.h>
```

6.507.1 Detailed Description

Definition at line 31 of file module_wrap_traits.h.

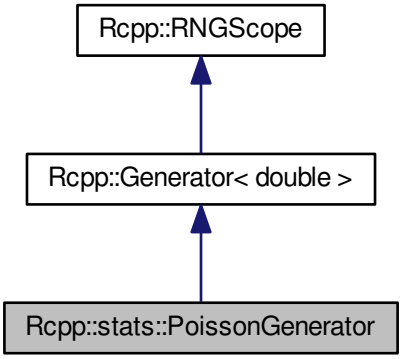
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

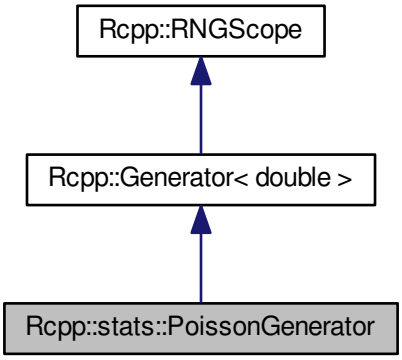
6.508 Rcpp::stats::PoissonGenerator Class Reference

`#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

6.508.1 Detailed Description

Definition at line 28 of file rpois.h.

6.508.2 Constructor & Destructor Documentation

6.508.2.1 Rcpp::stats::PoissonGenerator::PoissonGenerator (double *mu_*) [inline]

Definition at line 30 of file rpois.h.

6.508.3 Member Function Documentation

6.508.3.1 double Rcpp::stats::PoissonGenerator::operator() ( ) const [inline]

Definition at line 31 of file rpois.h.

References mu.

6.508.4 Member Data Documentation

6.508.4.1 double Rcpp::stats::PoissonGenerator::mu [private]

Definition at line 33 of file rpois.h.

Referenced by operator()().

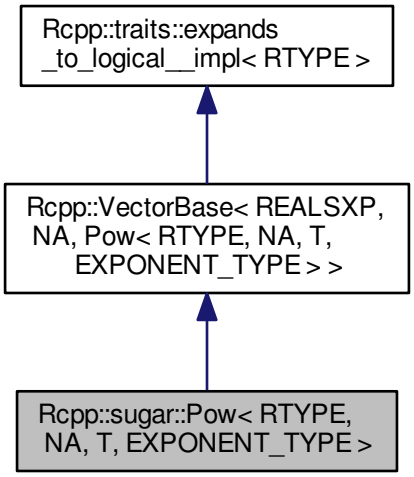
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rpois.h

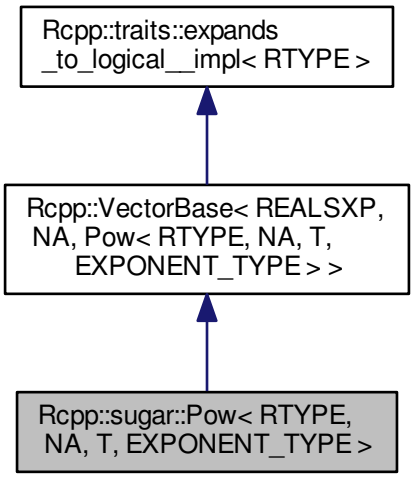
6.509 Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > Class Template Reference

`#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

6.509.1 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.

6.509.2 Member Typedef Documentation

6.509.2.1 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.

6.509.3 Constructor & Destructor Documentation

6.509.3.1 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.

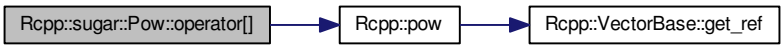
6.509.4 Member Function Documentation

6.509.4.1 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:



6.509.4.2 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.

6.509.5 Member Data Documentation

6.509.5.1 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.

6.509.5.2 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< INTS↩ XP, NA, T, EXPONENT_TYPE >::operator[ ](), and Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >↩ ::operator[ ]().

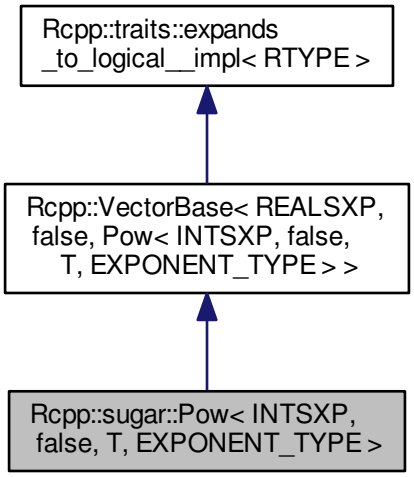
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pow.h

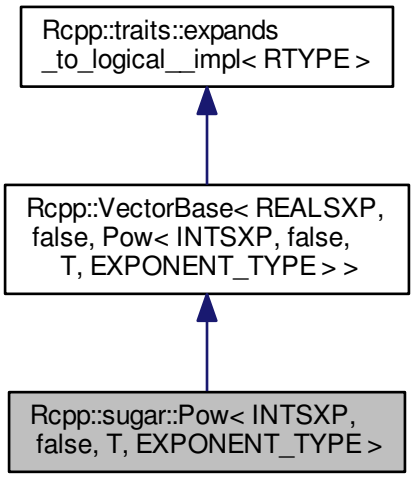
6.510 Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE > Class Template Reference

`#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

6.510.1 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.

6.510.2 Constructor & Destructor Documentation

6.510.2.1 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.

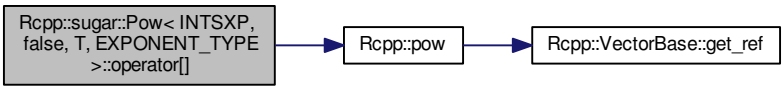
6.510.3 Member Function Documentation

6.510.3.1 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:



6.510.3.2 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.

6.510.4 Member Data Documentation

6.510.4.1 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.

6.510.4.2 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.

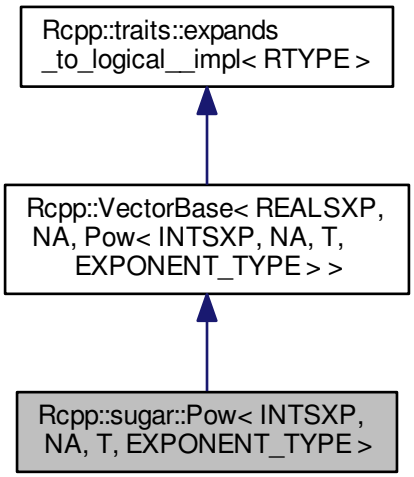
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pow.h

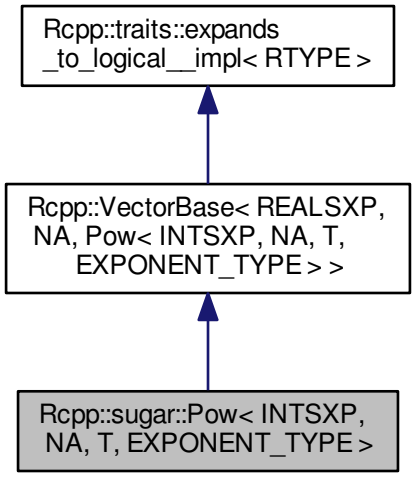
6.511 Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE > Class Template Reference

`#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

6.511.1 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.

6.511.2 Constructor & Destructor Documentation

6.511.2.1 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.

6.511.3 Member Function Documentation

6.511.3.1 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:

6.511.3.2 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.

6.511.4 Member Data Documentation

6.511.4.1 template<bool NA, typename T , typename EXPONENT_TYPE > const T& Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::object [private]

Definition at line 57 of file pow.h.

6.511.4.2 template<bool NA, typename T , typename EXPONENT_TYPE > EXPONENT_TYPE Rcpp::sugar::Pow< INTSXP, 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

6.512 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

6.512.1 Detailed Description

template<typename CLASS>
class Rcpp::PreserveStorage< CLASS >

Definition at line 7 of file PreserveStorage.h.

6.512.2 Constructor & Destructor Documentation

6.512.2.1 template<typename CLASS> Rcpp::PreserveStorage< CLASS >::PreserveStorage ( ) [inline]

Definition at line 10 of file PreserveStorage.h.

6.512.2.2 template<typename CLASS> Rcpp::PreserveStorage< CLASS >::~PreserveStorage ( ) [inline]

Definition at line 12 of file PreserveStorage.h.

6.512.3 Member Function Documentation

6.512.3.1 template<typename CLASS> template<typename T > T& Rcpp::PreserveStorage< CLASS >::copy__ (const T & *other*) [inline]

Definition at line 36 of file PreserveStorage.h.

6.512.3.2 template<typename CLASS> SEXP Rcpp::PreserveStorage< CLASS >::get__ ( ) const [inline]

Definition at line 25 of file PreserveStorage.h.

6.512.3.3 template<typename CLASS> bool Rcpp::PreserveStorage< CLASS >::inherits (const char * *clazz*) const [inline]

Definition at line 43 of file PreserveStorage.h.

6.512.3.4 template<typename CLASS> SEXP Rcpp::PreserveStorage< CLASS >::invalidate__ ( ) [inline]

Definition at line 29 of file PreserveStorage.h.

6.512.3.5 template<typename CLASS> Rcpp::PreserveStorage< CLASS >::operator SEXP ( ) const [inline]

Definition at line 47 of file PreserveStorage.h.

6.512.3.6 template<typename CLASS> void Rcpp::PreserveStorage< CLASS >::set__ (SEXP *x*) [inline]

Definition at line 17 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::copy__().

6.512.4 Member Data Documentation

6.512.4.1 template<typename CLASS> SEXP Rcpp::PreserveStorage< CLASS >::data [private]

Definition at line 50 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::get__(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::inherits(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::invalidate__(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::operator SEXP(), Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::set__(), and Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::~PreserveStorage().

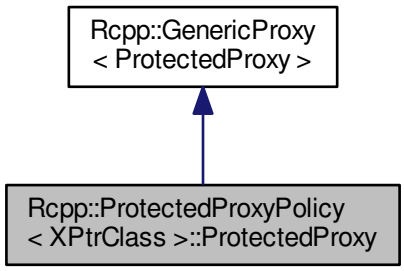
The documentation for this class was generated from the following file:

- inst/include/Rcpp/storage/PreserveStorage.h

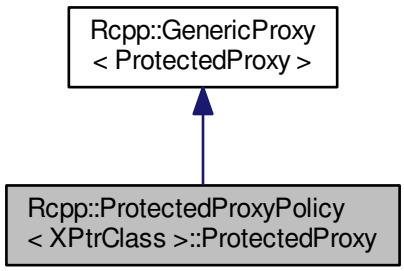
6.513 Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy Class Reference

`#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

6.513.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy

Definition at line 29 of file ProtectedProxy.h.

6.513.2 Constructor & Destructor Documentation

6.513.2.1 template<class XPtrClass> Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::ProtectedProxy (XPtrClass & *xp_*) [inline]

Definition at line 31 of file ProtectedProxy.h.

Referenced by Rcpp::ProtectedProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::prot().

6.513.3 Member Function Documentation

6.513.3.1 template<class XPtrClass> SEXP Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get ( ) const [inline], [private]

Definition at line 51 of file ProtectedProxy.h.

6.513.3.2 template<class XPtrClass> Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator SEXP ( ) const [inline]

Definition at line 44 of file ProtectedProxy.h.

6.513.3.3 template<class XPtrClass> template<typename U > Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator U ( ) const [inline]

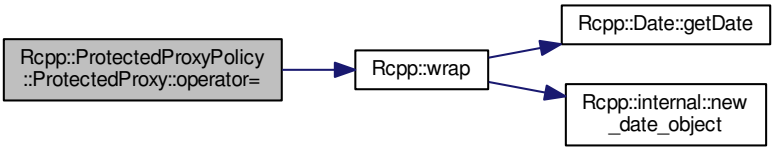
Definition at line 40 of file ProtectedProxy.h.

6.513.3.4 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::wrap().

Here is the call graph for this function:



6.513.3.5 template<class XPtrClass> void Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set (SEXP *x*) [inline], [private]

Definition at line 55 of file ProtectedProxy.h.

6.513.4 Member Data Documentation

6.513.4.1 template<class XPtrClass> XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp [private]

Definition at line 49 of file ProtectedProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

6.514 Rcpp::ProtectedProxyPolicy< XPtrClass > Class Template Reference

`#include <ProtectedProxy.h>`

Classes

- class const_ProtectedProxy
- class ProtectedProxy

Public Member Functions

- ProtectedProxy prot ()
- const_ProtectedProxy prot () const

6.514.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >

Definition at line 26 of file ProtectedProxy.h.

6.514.2 Member Function Documentation

6.514.2.1 template<class XPtrClass> ProtectedProxy Rcpp::ProtectedProxyPolicy< XPtrClass >::prot ( ) [inline]

Definition at line 83 of file ProtectedProxy.h.

6.514.2.2 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

6.515 Rcpp::traits::proxy_based_const_iterator< RTYPE > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_const_proxy< RTYPE >::type > type

6.515.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::proxy_based_const_iterator< RTYPE >

Definition at line 253 of file proxy.h.

6.515.2 Member Typedef Documentation

6.515.2.1 template<int RTYPE> typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_const_proxy<RTYPE>::type > Rcpp::traits::proxy_based_const_iterator< RTYPE >::type

Definition at line 254 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.516 Rcpp::traits::proxy_based_iterator< RTYPE > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_proxy< RTYPE >::type > type

6.516.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::proxy_based_iterator< RTYPE >

Definition at line 246 of file proxy.h.

6.516.2 Member Typedef Documentation

6.516.2.1 template<int RTYPE> typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_proxy<RTYPE>::type > Rcpp::traits::proxy_based_iterator< RTYPE >::type

Definition at line 247 of file proxy.h.

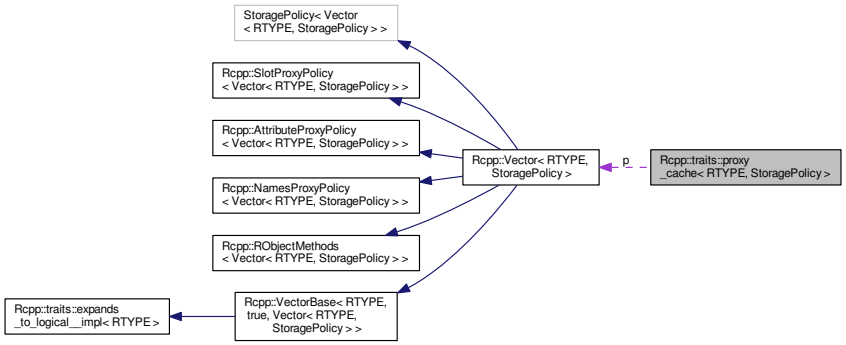
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.517 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 >::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

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

6.517.1 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.

6.517.2 Member Typedef Documentation

6.517.2.1 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef r_vector_const_iterator<RTYPE>::type Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::const_iterator

Definition at line 59 of file traits.h.

6.517.2.2 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef r_vector_const_proxy<RTYPE>::type Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::const_proxy

Definition at line 61 of file traits.h.

6.517.2.3 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef r_vector_iterator<RTYPE>::type Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::iterator

Definition at line 58 of file traits.h.

6.517.2.4 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef r_vector_proxy<RTYPE>::type Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::proxy

Definition at line 60 of file traits.h.

6.517.2.5 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.

6.517.3 Constructor & Destructor Documentation

6.517.3.1 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.

6.517.3.2 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.

6.517.4 Member Function Documentation

6.517.4.1 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.

6.517.4.2 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.

6.517.4.3 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.

6.517.4.4 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.

6.517.4.5 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.

6.517.4.6 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.

6.517.4.7 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.

6.517.5 Member Data Documentation

6.517.5.1 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.

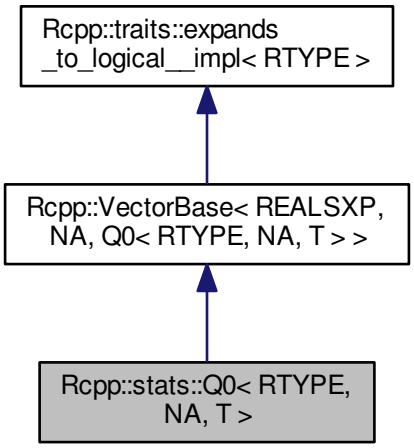
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/traits.h

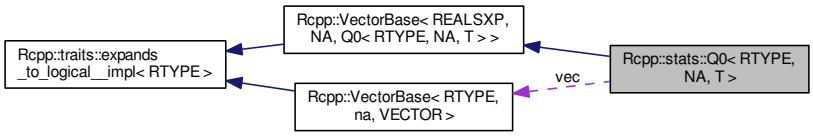
6.518 Rcpp::stats::Q0< RTYPE, NA, T > Class Template Reference

`#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

6.518.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q0< RTYPE, NA, T >

Definition at line 221 of file dpq.h.

6.518.2 Member Typedef Documentation

6.518.2.1 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.

6.518.2.2 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.

6.518.3 Constructor & Destructor Documentation

6.518.3.1 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.

6.518.4 Member Function Documentation

6.518.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.518.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.518.5 Member Data Documentation

6.518.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q0< RTYPE, NA, T >::log [private]

Definition at line 239 of file dpq.h.

6.518.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q0< RTYPE, NA, T >::lower [private]

Definition at line 239 of file dpq.h.

6.518.5.3 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::Q0< RTYPE, NA, T >::ptr [private]

Definition at line 237 of file dpq.h.

6.518.5.4 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.

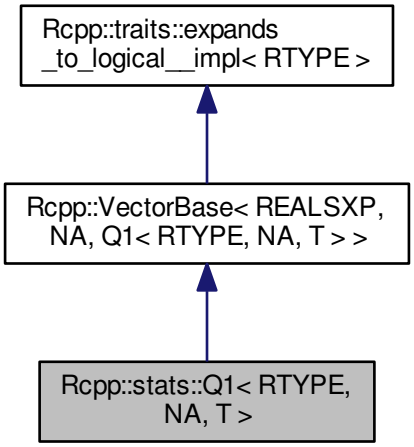
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

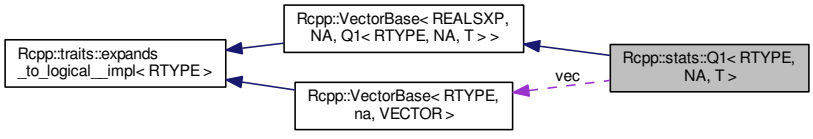
6.519 Rcpp::stats::Q1< RTYPE, NA, T > Class Template Reference

`#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

6.519.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q1< RTYPE, NA, T >

Definition at line 244 of file dpq.h.

6.519.2 Member Typedef Documentation

6.519.2.1 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.

6.519.2.2 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.

6.519.3 Constructor & Destructor Documentation

6.519.3.1 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.

6.519.4 Member Function Documentation

6.519.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.519.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.519.5 Member Data Documentation

6.519.5.1 template<int RTYPE, bool NA, typename T> int Rcpp::stats::Q1< RTYPE, NA, T >::log [private]

Definition at line 263 of file dpq.h.

6.519.5.2 template<int RTYPE, bool NA, typename T> int Rcpp::stats::Q1< RTYPE, NA, T >::lower [private]

Definition at line 263 of file dpq.h.

6.519.5.3 template<int RTYPE, bool NA, typename T> double Rcpp::stats::Q1< RTYPE, NA, T >::p0 [private]

Definition at line 262 of file dpq.h.

6.519.5.4 template<int RTYPE, bool NA, typename T> FunPtr Rcpp::stats::Q1< RTYPE, NA, T >::ptr [private]

Definition at line 260 of file dpq.h.

6.519.5.5 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.

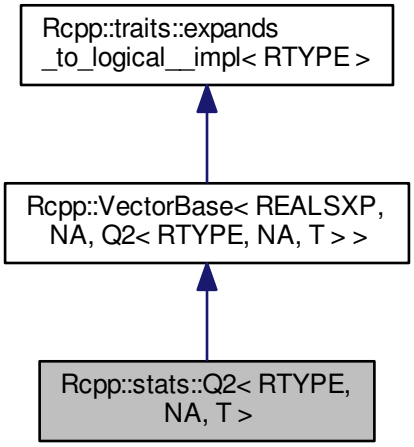
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

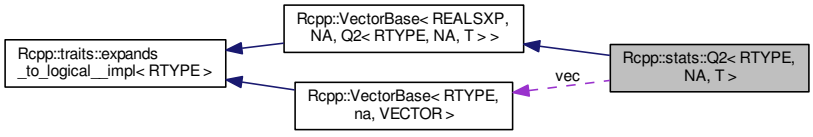
6.520 Rcpp::stats::Q2< RTYPE, NA, T > Class Template Reference

`#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

6.520.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q2< RTYPE, NA, T >

Definition at line 268 of file dpq.h.

6.520.2 Member Typedef Documentation

6.520.2.1 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.

6.520.2.2 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.

6.520.3 Constructor & Destructor Documentation

6.520.3.1 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.

6.520.4 Member Function Documentation

6.520.4.1 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::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.520.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:

6.520.5 Member Data Documentation

6.520.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q2< RTYPE, NA, T >::log [private]

Definition at line 287 of file dpq.h.

6.520.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q2< RTYPE, NA, T >::lower [private]

Definition at line 287 of file dpq.h.

6.520.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::Q2< RTYPE, NA, T >::p0 [private]

Definition at line 286 of file dpq.h.

6.520.5.4 template<int RTYPE, bool NA, typename T > double Rcpp::stats::Q2< RTYPE, NA, T >::p1 [private]

Definition at line 286 of file dpq.h.

6.520.5.5 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::Q2< RTYPE, NA, T >::ptr [private]

Definition at line 284 of file dpq.h.

6.520.5.6 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.

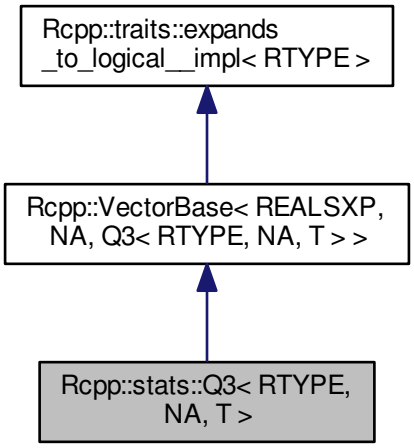
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

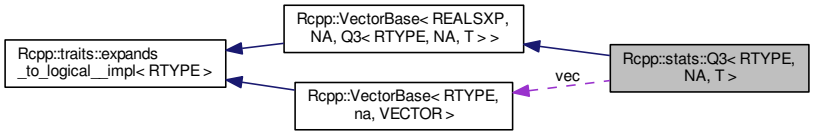
6.521 Rcpp::stats::Q3< RTYPE, NA, T > Class Template Reference

`#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

6.521.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q3< RTYPE, NA, T >

Definition at line 292 of file dpq.h.

6.521.2 Member Typedef Documentation

6.521.2.1 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.

6.521.2.2 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.

6.521.3 Constructor & Destructor Documentation

6.521.3.1 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.

6.521.4 Member Function Documentation

6.521.4.1 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.

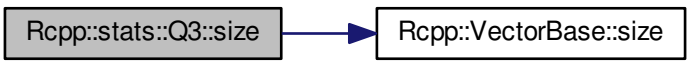
References Rcpp::stats::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

6.521.4.2 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::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.521.5 Member Data Documentation

6.521.5.1 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q3< RTYPE, NA, T >::log [private]

Definition at line 311 of file dpq.h.

6.521.5.2 template<int RTYPE, bool NA, typename T > int Rcpp::stats::Q3< RTYPE, NA, T >::lower [private]

Definition at line 311 of file dpq.h.

6.521.5.3 template<int RTYPE, bool NA, typename T > double Rcpp::stats::Q3< RTYPE, NA, T >::p0 [private]

Definition at line 310 of file dpq.h.

6.521.5.4 template<int RTYPE, bool NA, typename T > double Rcpp::stats::Q3< RTYPE, NA, T >::p1 [private]

Definition at line 310 of file dpq.h.

6.521.5.5 template<int RTYPE, bool NA, typename T > double Rcpp::stats::Q3< RTYPE, NA, T >::p2 [private]

Definition at line 310 of file dpq.h.

6.521.5.6 template<int RTYPE, bool NA, typename T > FunPtr Rcpp::stats::Q3< RTYPE, NA, T >::ptr [private]

Definition at line 308 of file dpq.h.

6.521.5.7 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

6.522 Rcpp::traits::expands_to_logical_impl< LGLSXP >::r_expands_to_logical Struct Reference

`#include <expands_to_logical.h>`

6.522.1 Detailed Description

template<>
struct Rcpp::traits::expands_to_logical__impl< LGLSXP >::r_expands_to_logical

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

6.523 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface Struct Reference

`#include <MatrixBase.h>`

6.523.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface

Definition at line 34 of file MatrixBase.h.

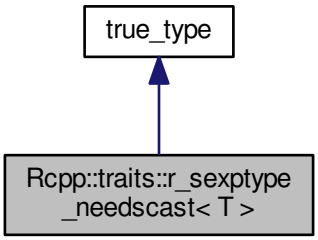
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

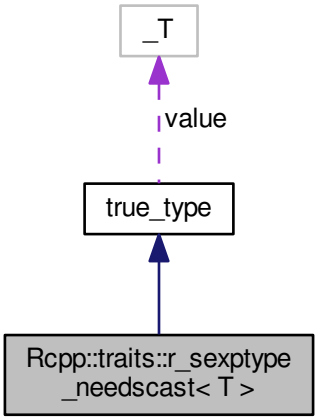
6.524 Rcpp::traits::r_sexptype_needscast< T > Struct Template Reference

`#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

6.524.1 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.

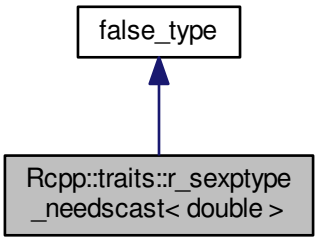
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

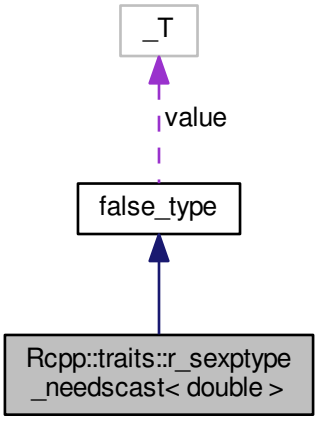
6.525 Rcpp::traits::r_sexptype_needscast< double > Struct Template Reference

`#include <r_sexptype_traits.h>`

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< double >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< double >:



Additional Inherited Members

6.525.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_needscast< double >

Definition at line 71 of file r_sexptype_traits.h.

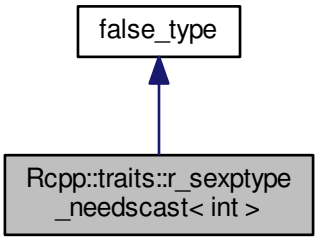
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

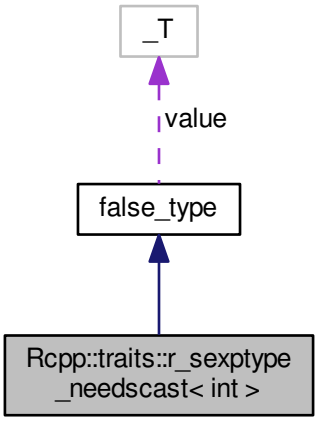
6.526 Rcpp::traits::r_sexptype_needscast< int > Struct Template Reference

`#include <r_sexptype_traits.h>`

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< int >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< int >:



Additional Inherited Members

6.526.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_needscast< int >

Definition at line 70 of file r_sexptype_traits.h.

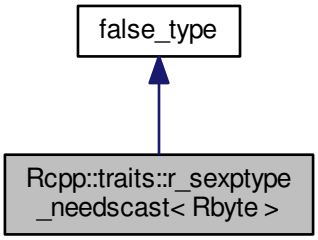
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

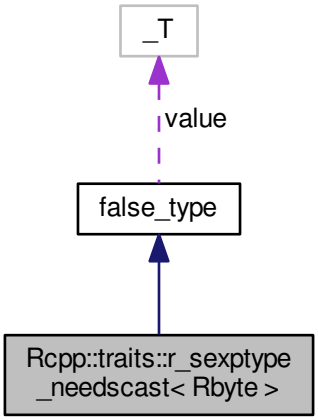
6.527 Rcpp::traits::r_sexptype_needscast< Rbyte > Struct Template Reference

`#include <r_sexptype_traits.h>`

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< Rbyte >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< Rbyte >:



Additional Inherited Members

6.527.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_needscast< Rbyte >

Definition at line 73 of file r_sexptype_traits.h.

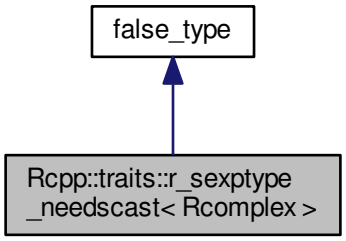
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

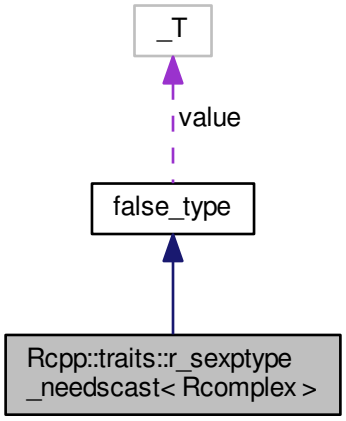
6.528 Rcpp::traits::r_sexptype_needscast< Rcomplex > Struct Template Reference

`#include <r_sexptype_traits.h>`

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< Rcomplex >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< Rcomplex >:



Additional Inherited Members

6.528.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_needscast< Rcomplex >

Definition at line 72 of file r_sexptype_traits.h.

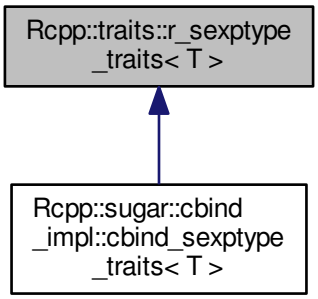
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.529 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 = VECSXP }

6.529.1 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 VECSXP (lists) unless it is specialized

Definition at line 33 of file r_sexptype_traits.h.

6.529.2 Member Enumeration Documentation

6.529.2.1 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

6.530 Rcpp::traits::r_sexptype_traits< bool > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = LGLSXP }

6.530.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< bool >

Definition at line 38 of file r_sexptype_traits.h.

6.530.2 Member Enumeration Documentation

6.530.2.1 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

6.531 Rcpp::traits::r_sexptype_traits< const double > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.531.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< const double >

Definition at line 37 of file r_sexptype_traits.h.

6.531.2 Member Enumeration Documentation

6.531.2.1 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

6.532 Rcpp::traits::r_sexptype_traits< const int > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = INTSXP }

6.532.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< const int >

Definition at line 35 of file r_sexptype_traits.h.

6.532.2 Member Enumeration Documentation

6.532.2.1 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

6.533 Rcpp::traits::r_sexptype_traits< double > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.533.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< double >

Definition at line 36 of file r_sexptype_traits.h.

6.533.2 Member Enumeration Documentation

6.533.2.1 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

6.534 Rcpp::traits::r_sexptype_traits< float > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.534.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< float >

Definition at line 45 of file r_sexptype_traits.h.

6.534.2 Member Enumeration Documentation

6.534.2.1 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

6.535 Rcpp::traits::r_sexptype_traits< int > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = INTSXP }

6.535.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< int >

Definition at line 34 of file r_sexptype_traits.h.

6.535.2 Member Enumeration Documentation

6.535.2.1 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

6.536 Rcpp::traits::r_sexptype_traits< long > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.536.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< long >

Definition at line 49 of file r_sexptype_traits.h.

6.536.2 Member Enumeration Documentation

6.536.2.1 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

6.537 Rcpp::traits::r_sexptype_traits< long double > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.537.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< long double >

Definition at line 55 of file r_sexptype_traits.h.

6.537.2 Member Enumeration Documentation

6.537.2.1 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

6.538 Rcpp::traits::r_sexptype_traits< Rbyte > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = RAWSXP }

6.538.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< Rbyte >

Definition at line 41 of file r_sexptype_traits.h.

6.538.2 Member Enumeration Documentation

6.538.2.1 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

6.539 Rcpp::traits::r_sexptype_traits< Rcomplex > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = CPLXSXP }

6.539.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< Rcomplex >

Definition at line 40 of file r_sexptype_traits.h.

6.539.2 Member Enumeration Documentation

6.539.2.1 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

6.540 Rcpp::traits::r_sexptype_traits< Rcpp::Date > Struct Template Reference

`#include <Date.h>`

Public Types

- enum { rtype = REALSXP }

6.540.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< Rcpp::Date >

Definition at line 37 of file Date.h.

6.540.2 Member Enumeration Documentation

6.540.2.1 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

6.541 Rcpp::traits::r_sexptype_traits< Rcpp::Datetime > Struct Template Reference

`#include <Datetime.h>`

Public Types

- enum { rtype = REALSXP }

6.541.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< Rcpp::Datetime >

Definition at line 37 of file Datetime.h.

6.541.2 Member Enumeration Documentation

6.541.2.1 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

6.542 Rcpp::traits::r_sexptype_traits< Rcpp::String > Struct Template Reference

`#include <String.h>`

Public Types

- enum { rtype = STRSXP }

6.542.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< Rcpp::String >

Definition at line 477 of file String.h.

6.542.2 Member Enumeration Documentation

6.542.2.1 anonymous enum

Enumerator

rtype

Definition at line 477 of file String.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/String.h

6.543 Rcpp::traits::r_sexptype_traits< short > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = INTSXP }

6.543.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< short >

Definition at line 59 of file r_sexptype_traits.h.

6.543.2 Member Enumeration Documentation

6.543.2.1 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

6.544 Rcpp::traits::r_sexptype_traits< std::complex< double > > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = CPLXSXP }

6.544.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< std::complex< double > >

Definition at line 63 of file r_sexptype_traits.h.

6.544.2 Member Enumeration Documentation

6.544.2.1 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

6.545 Rcpp::traits::r_sexptype_traits< std::complex< float > > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = CPLXSXP }

6.545.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< std::complex< float > >

Definition at line 64 of file r_sexptype_traits.h.

6.545.2 Member Enumeration Documentation

6.545.2.1 anonymous enum

Enumerator

rtype

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

6.546 Rcpp::traits::r_sexptype_traits< std::string > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = STRSXP }

6.546.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< std::string >

Definition at line 39 of file r_sexptype_traits.h.

6.546.2 Member Enumeration Documentation

6.546.2.1 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

6.547 Rcpp::traits::r_sexptype_traits< unsigned int > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.547.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< unsigned int >

Definition at line 44 of file r_sexptype_traits.h.

6.547.2 Member Enumeration Documentation

6.547.2.1 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

6.548 Rcpp::traits::r_sexptype_traits< unsigned long > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = REALSXP }

6.548.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< unsigned long >

Definition at line 50 of file r_sexptype_traits.h.

6.548.2 Member Enumeration Documentation

6.548.2.1 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

6.549 Rcpp::traits::r_sexptype_traits< unsigned short > Struct Template Reference

`#include <r_sexptype_traits.h>`

Public Types

- enum { rtype = INTSXP }

6.549.1 Detailed Description

template<>
struct Rcpp::traits::r_sexptype_traits< unsigned short >

Definition at line 60 of file r_sexptype_traits.h.

6.549.2 Member Enumeration Documentation

6.549.2.1 anonymous enum

Enumerator

rtype

Definition at line 60 of file r_sexptype_traits.h.

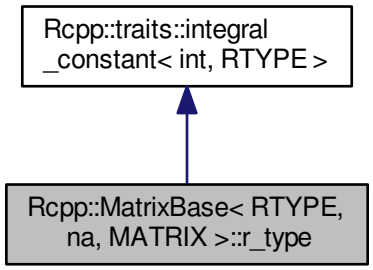
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

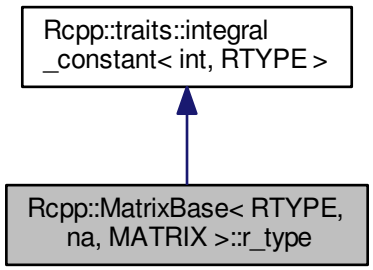
6.550 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type Struct Reference

`#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

6.550.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type

Definition at line 33 of file MatrixBase.h.

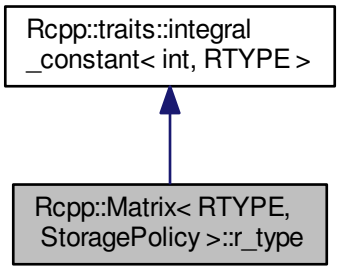
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

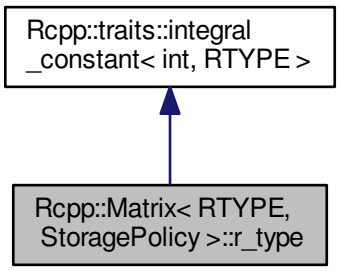
6.551 Rcpp::Matrix< RTYPE, StoragePolicy >::r_type Struct Reference

`#include <Matrix.h>`

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r_type:



Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r_type:



Additional Inherited Members

6.551.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::Matrix< RTYPE, StoragePolicy >::r_type

Definition at line 32 of file Matrix.h.

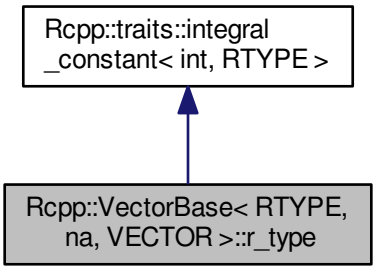
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/Matrix.h

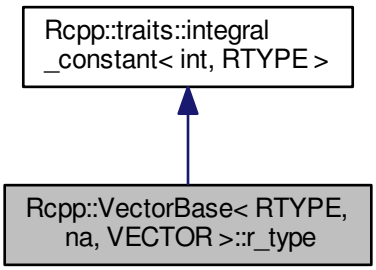
6.552 Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type Struct Reference

`#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

6.552.1 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

6.553 Rcpp::traits::r_type_enum_tag Struct Reference

`#include <r_type_traits.h>`

6.553.1 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

6.554 Rcpp::traits::r_type_generic_tag Struct Reference

`#include <r_type_traits.h>`

6.554.1 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

6.555 Rcpp::traits::r_type_module_object_const_pointer_tag Struct Reference

`#include <r_type_traits.h>`

6.555.1 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

6.556 Rcpp::traits::r_type_module_object_const_reference_tag Struct Reference

`#include <r_type_traits.h>`

6.556.1 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

6.557 Rcpp::traits::r_type_module_object_pointer_tag Struct Reference

`#include <r_type_traits.h>`

6.557.1 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

6.558 Rcpp::traits::r_type_module_object_reference_tag Struct Reference

`#include <r_type_traits.h>`

6.558.1 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

6.559 Rcpp::traits::r_type_module_object_tag Struct Reference

`#include <r_type_traits.h>`

6.559.1 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

6.560 Rcpp::traits::r_type_pair_tag Struct Reference

`#include <r_type_traits.h>`

6.560.1 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

6.561 Rcpp::traits::r_type_pairstring_generic_tag Struct Reference

`#include <r_type_traits.h>`

6.561.1 Detailed Description

Indentifies 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

6.562 Rcpp::traits::r_type_pairstring_primitive_tag Struct Reference

`#include <r_type_traits.h>`

6.562.1 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

6.563 Rcpp::traits::r_type_pairstring_string_tag Struct Reference

`#include <r_type_traits.h>`

6.563.1 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

6.564 Rcpp::traits::r_type_primitive_tag Struct Reference

`#include <r_type_traits.h>`

6.564.1 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

6.565 Rcpp::traits::r_type_RcppString_tag Struct Reference

`#include <r_type_traits.h>`

6.565.1 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

6.566 Rcpp::traits::r_type_string_tag Struct Reference

`#include <r_type_traits.h>`

6.566.1 Detailed Description

Identifies that the associated type can be implicitely 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

6.567 Rcpp::traits::r_type_traits< T > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_generic_tag r_category

6.567.1 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.

6.567.2 Member Typedef Documentation

6.567.2.1 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

6.568 Rcpp::traits::r_type_traits< bool > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.568.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< bool >

Definition at line 141 of file r_type_traits.h.

6.568.2 Member Typedef Documentation

6.568.2.1 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

6.569 Rcpp::traits::r_type_traits< char > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_string_tag r_category

6.569.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< char >

Definition at line 144 of file r_type_traits.h.

6.569.2 Member Typedef Documentation

6.569.2.1 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

6.570 Rcpp::traits::r_type_traits< const char * > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_string_tag r_category

6.570.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< const char * >

Definition at line 149 of file r_type_traits.h.

6.570.2 Member Typedef Documentation

6.570.2.1 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

6.571 Rcpp::traits::r_type_traits< const double > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.571.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< const double >

Definition at line 138 of file r_type_traits.h.

6.571.2 Member Typedef Documentation

6.571.2.1 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

6.572 Rcpp::traits::r_type_traits< const int > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.572.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< const int >

Definition at line 136 of file r_type_traits.h.

6.572.2 Member Typedef Documentation

6.572.2.1 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

6.573 Rcpp::traits::r_type_traits< const wchar_t * > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_string_tag r_category

6.573.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< const wchar_t * >

Definition at line 150 of file r_type_traits.h.

6.573.2 Member Typedef Documentation

6.573.2.1 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

6.574 Rcpp::traits::r_type_traits< double > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.574.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< double >

Definition at line 137 of file r_type_traits.h.

6.574.2 Member Typedef Documentation

6.574.2.1 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

6.575 Rcpp::traits::r_type_traits< float > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.575.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< float >

Definition at line 148 of file r_type_traits.h.

6.575.2 Member Typedef Documentation

6.575.2.1 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

6.576 Rcpp::traits::r_type_traits< int > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.576.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< int >

Definition at line 135 of file r_type_traits.h.

6.576.2 Member Typedef Documentation

6.576.2.1 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

6.577 Rcpp::traits::r_type_traits< long > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.577.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< long >

Definition at line 153 of file r_type_traits.h.

6.577.2 Member Typedef Documentation

6.577.2.1 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

6.578 Rcpp::traits::r_type_traits< long double > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.578.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< long double >

Definition at line 161 of file r_type_traits.h.

6.578.2 Member Typedef Documentation

6.578.2.1 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

6.579 Rcpp::traits::r_type_traits< Rbyte > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.579.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< Rbyte >

Definition at line 139 of file r_type_traits.h.

6.579.2 Member Typedef Documentation

6.579.2.1 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

6.580 Rcpp::traits::r_type_traits< Rcomplex > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.580.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< Rcomplex >

Definition at line 140 of file r_type_traits.h.

6.580.2 Member Typedef Documentation

6.580.2.1 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

6.581 Rcpp::traits::r_type_traits< Rcpp::Date > Struct Template Reference

`#include <Date.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.581.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< Rcpp::Date >

Definition at line 31 of file Date.h.

6.581.2 Member Typedef Documentation

6.581.2.1 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

6.582 Rcpp::traits::r_type_traits< Rcpp::Datetime > Struct Template Reference

`#include <Datetime.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.582.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< Rcpp::Datetime >

Definition at line 31 of file Datetime.h.

6.582.2 Member Typedef Documentation

6.582.2.1 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

6.583 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

6.583.1 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.

6.583.2 Member Typedef Documentation

6.583.2.1 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

6.584 Rcpp::traits::r_type_traits< Rcpp::String > Struct Template Reference

`#include <String.h>`

Public Types

- typedef r_type_RcppString_tag r_category

6.584.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< Rcpp::String >

Definition at line 476 of file String.h.

6.584.2 Member Typedef Documentation

6.584.2.1 typedef r_type_RcppString_tag Rcpp::traits::r_type_traits< Rcpp::String >::r_category

Definition at line 476 of file String.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/String.h

6.585 Rcpp::traits::r_type_traits< short > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.585.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< short >

Definition at line 165 of file r_type_traits.h.

6.585.2 Member Typedef Documentation

6.585.2.1 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

6.586 Rcpp::traits::r_type_traits< std::complex< double > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.586.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::complex< double > >

Definition at line 173 of file r_type_traits.h.

6.586.2 Member Typedef Documentation

6.586.2.1 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

6.587 Rcpp::traits::r_type_traits< std::complex< float > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.587.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::complex< float > >

Definition at line 177 of file r_type_traits.h.

6.587.2 Member Typedef Documentation

6.587.2.1 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

6.588 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

6.588.1 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.

6.588.2 Member Typedef Documentation

6.588.2.1 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

6.589 Rcpp::traits::r_type_traits< std::pair< const std::string, bool > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.589.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, bool > >

Definition at line 126 of file r_type_traits.h.

6.589.2 Member Typedef Documentation

6.589.2.1 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

6.590 Rcpp::traits::r_type_traits< std::pair< const std::string, char > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_string_tag r_category

6.590.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, char > >

Definition at line 129 of file r_type_traits.h.

6.590.2 Member Typedef Documentation

6.590.2.1 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

6.591 Rcpp::traits::r_type_traits< std::pair< const std::string, const int > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.591.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, const int > >

Definition at line 122 of file r_type_traits.h.

6.591.2 Member Typedef Documentation

6.591.2.1 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

6.592 Rcpp::traits::r_type_traits< std::pair< const std::string, double > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.592.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, double > >

Definition at line 123 of file r_type_traits.h.

6.592.2 Member Typedef Documentation

6.592.2.1 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

6.593 Rcpp::traits::r_type_traits< std::pair< const std::string, float > > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.593.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, float > >

Definition at line 133 of file r_type_traits.h.

6.593.2 Member Typedef Documentation

6.593.2.1 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

6.594 Rcpp::traits::r_type_traits< std::pair< const std::string, int > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.594.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, int > >

Definition at line 121 of file r_type_traits.h.

6.594.2 Member Typedef Documentation

6.594.2.1 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

6.595 Rcpp::traits::r_type_traits< std::pair< const std::string, long > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.595.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, long > >

Definition at line 154 of file r_type_traits.h.

6.595.2 Member Typedef Documentation

6.595.2.1 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

6.596 Rcpp::traits::r_type_traits< std::pair< const std::string, long double > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.596.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, long double > >

Definition at line 162 of file r_type_traits.h.

6.596.2 Member Typedef Documentation

6.596.2.1 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

6.597 Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.597.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > >

Definition at line 124 of file r_type_traits.h.

6.597.2 Member Typedef Documentation

6.597.2.1 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

6.598 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.598.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > >

Definition at line 125 of file r_type_traits.h.

6.598.2 Member Typedef Documentation

6.598.2.1 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

6.599 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > > Struct Template Reference

`#include <Date.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.599.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > >

Definition at line 34 of file Date.h.

6.599.2 Member Typedef Documentation

6.599.2.1 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

6.600 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > > Struct Template Reference

`#include <Datetime.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.600.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > >

Definition at line 34 of file Datetime.h.

6.600.2 Member Typedef Documentation

6.600.2.1 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

6.601 Rcpp::traits::r_type_traits< std::pair< const std::string, short > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.601.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, short > >

Definition at line 166 of file r_type_traits.h.

6.601.2 Member Typedef Documentation

6.601.2.1 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

6.602 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.602.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > >

Definition at line 174 of file r_type_traits.h.

6.602.2 Member Typedef Documentation

6.602.2.1 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

6.603 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.603.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > >

Definition at line 178 of file r_type_traits.h.

6.603.2 Member Typedef Documentation

6.603.2.1 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

6.604 Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_string_tag r_category

6.604.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > >

Definition at line 127 of file r_type_traits.h.

6.604.2 Member Typedef Documentation

6.604.2.1 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

6.605 Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_string_tag r_category

6.605.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > >

Definition at line 128 of file r_type_traits.h.

6.605.2 Member Typedef Documentation

6.605.2.1 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

6.606 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

6.606.1 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.

6.606.2 Member Typedef Documentation

6.606.2.1 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

6.607 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned int > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.607.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned int > >

Definition at line 132 of file r_type_traits.h.

6.607.2 Member Typedef Documentation

6.607.2.1 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

6.608 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned long > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.608.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned long > >

Definition at line 158 of file r_type_traits.h.

6.608.2 Member Typedef Documentation

6.608.2.1 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

6.609 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned short > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.609.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned short > >

Definition at line 170 of file r_type_traits.h.

6.609.2 Member Typedef Documentation

6.609.2.1 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

6.610 Rcpp::traits::r_type_traits< std::pair< const std::string, wchar_t > > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_pairstring_string_tag r_category

6.610.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, wchar_t > >

Definition at line 130 of file r_type_traits.h.

6.610.2 Member Typedef Documentation

6.610.2.1 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

6.611 Rcpp::traits::r_type_traits< std::string > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.611.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::string >

Definition at line 142 of file r_type_traits.h.

6.611.2 Member Typedef Documentation

6.611.2.1 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

6.612 Rcpp::traits::r_type_traits< std::wstring > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.612.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< std::wstring >

Definition at line 143 of file r_type_traits.h.

6.612.2 Member Typedef Documentation

6.612.2.1 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

6.613 Rcpp::traits::r_type_traits< unsigned int > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.613.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< unsigned int >

Definition at line 147 of file r_type_traits.h.

6.613.2 Member Typedef Documentation

6.613.2.1 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

6.614 Rcpp::traits::r_type_traits< unsigned long > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.614.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< unsigned long >

Definition at line 157 of file r_type_traits.h.

6.614.2 Member Typedef Documentation

6.614.2.1 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

6.615 Rcpp::traits::r_type_traits< unsigned short > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_primitive_tag r_category

6.615.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< unsigned short >

Definition at line 169 of file r_type_traits.h.

6.615.2 Member Typedef Documentation

6.615.2.1 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

6.616 Rcpp::traits::r_type_traits< wchar_t > Struct Template Reference

`#include <r_type_traits.h>`

Public Types

- typedef r_type_string_tag r_category

6.616.1 Detailed Description

template<>
struct Rcpp::traits::r_type_traits< wchar_t >

Definition at line 145 of file r_type_traits.h.

6.616.2 Member Typedef Documentation

6.616.2.1 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

6.617 Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy > Class Template Reference

`#include <00_forward_proxy.h>`

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

6.617.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >

Definition at line 39 of file 00_forward_proxy.h.

6.617.2 Member Typedef Documentation

6.617.2.1 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.

6.617.2.2 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.

6.617.2.3 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.

6.617.2.4 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.

6.617.2.5 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.

6.617.2.6 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.

6.617.3 Constructor & Destructor Documentation

6.617.3.1 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.

6.617.4 Member Function Documentation

6.617.4.1 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.

Referenced by Rcpp::Vector< INTSXP >::begin(), Rcpp::Vector< INTSXP >::end(), Rcpp::Vector< INTSXP >::erase(), and Rcpp::Vector< INTSXP >::insert().

6.617.4.2 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.

Referenced by Rcpp::Vector< INTSXP >::begin(), and Rcpp::Vector< INTSXP >::end().

6.617.4.3 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.

Referenced by Rcpp::Vector< INTSXP >::at(), Rcpp::Vector< INTSXP >::operator()(), and Rcpp::Vector< INTSXP >::operator[]().

6.617.4.4 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.

6.617.4.5 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.

6.617.4.6 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.

6.617.4.7 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.

Referenced by Rcpp::Vector< INTSXP >::update().

6.617.5 Member Data Documentation

6.617.5.1 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

6.618 Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef r_vector_cache< RTYPE, StoragePolicy > type

6.618.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >

Definition at line 38 of file 00_forward_proxy.h.

6.618.2 Member Typedef Documentation

6.618.2.1 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

6.619 Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy > Struct Template Reference

`#include <traits.h>`

Public Types

- typedef proxy_cache< EXPRSXP, StoragePolicy > type

6.619.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy >

Definition at line 94 of file traits.h.

6.619.2 Member Typedef Documentation

6.619.2.1 template<template< class > class StoragePolicy> typedef proxy_cache<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy >::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

6.620 Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy > Struct Template Reference

`#include <traits.h>`

Public Types

- typedef proxy_cache< STRSXP, StoragePolicy > type

6.620.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >

Definition at line 98 of file traits.h.

6.620.2 Member Typedef Documentation

6.620.2.1 template<template< class > class StoragePolicy> typedef proxy_cache<STRSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >::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

6.621 Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy > Struct Template Reference

`#include <traits.h>`

Public Types

- typedef proxy_cache< VECSXP, StoragePolicy > type

6.621.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >

Definition at line 90 of file traits.h.

6.621.2 Member Typedef Documentation

6.621.2.1 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

6.622 Rcpp::traits::r_vector_const_iterator< RTYPE > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef const storage_type< RTYPE >::type * type

6.622.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_const_iterator< RTYPE >

Definition at line 46 of file 00_forward_proxy.h.

6.622.2 Member Typedef Documentation

6.622.2.1 template<int RTYPE> typedef const storage_type<RTYPE>::type* Rcpp::traits::r_vector_const_iterator< RTYPE >::type

Definition at line 243 of file proxy.h.

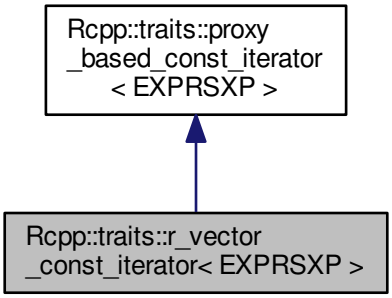
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

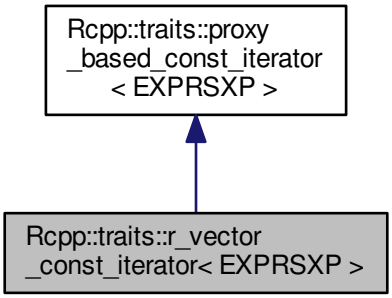
6.623 Rcpp::traits::r_vector_const_iterator< EXPRSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< EXPRSXP >:



Collaboration diagram for Rcpp::traits::r_vector_const_iterator< EXPRSXP >:



Additional Inherited Members

6.623.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_iterator< EXPRSXP >

Definition at line 257 of file proxy.h.

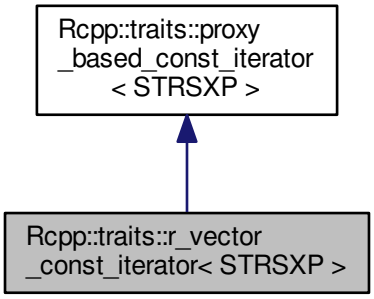
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

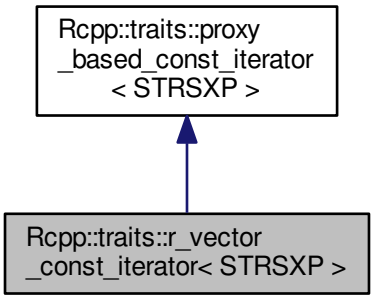
6.624 Rcpp::traits::r_vector_const_iterator< STRSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< STRSXP >:



Collaboration diagram for Rcpp::traits::r_vector_const_iterator< STRSXP >:



Additional Inherited Members

6.624.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_iterator< STRSXP >

Definition at line 258 of file proxy.h.

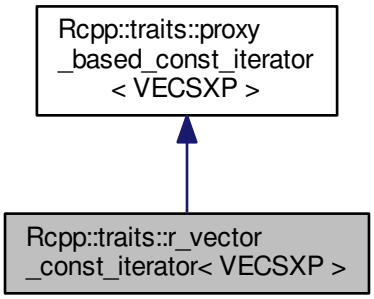
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

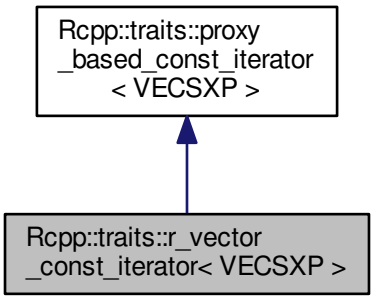
6.625 Rcpp::traits::r_vector_const_iterator< VECSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< VECSXP >:



Collaboration diagram for Rcpp::traits::r_vector_const_iterator< VECSXP >:



Additional Inherited Members

6.625.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_iterator< VECSXP >

Definition at line 256 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.626 Rcpp::traits::r_vector_const_proxy< RTYPE > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef const storage_type< RTYPE >::type & type

6.626.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_const_proxy< RTYPE >

Definition at line 43 of file 00_forward_proxy.h.

6.626.2 Member Typedef Documentation

6.626.2.1 template<int RTYPE> typedef const storage_type<RTYPE>::type& Rcpp::traits::r_vector_const_proxy< RTYPE >::type

Definition at line 225 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

6.627 Rcpp::traits::r_vector_const_proxy< EXPRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::const_generic_proxy< EXPRSXP > type

6.627.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_proxy< EXPRSXP >

Definition at line 233 of file proxy.h.

6.627.2 Member Typedef Documentation

6.627.2.1 typedef ::Rcpp::internal::const_generic_proxy<EXPRSXP> Rcpp::traits::r_vector_const_proxy< EXPRSXP >::type

Definition at line 234 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.628 Rcpp::traits::r_vector_const_proxy< STRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::const_string_proxy< STRSXP > type

6.628.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_proxy< STRSXP >

Definition at line 227 of file proxy.h.

6.628.2 Member Typedef Documentation

6.628.2.1 typedef ::Rcpp::internal::const_string_proxy<STRSXP> Rcpp::traits::r_vector_const_proxy< STRSXP >::type

Definition at line 228 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.629 Rcpp::traits::r_vector_const_proxy< VECSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::const_generic_proxy< VECSXP > type

6.629.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_const_proxy< VECSXP >

Definition at line 230 of file proxy.h.

6.629.2 Member Typedef Documentation

6.629.2.1 typedef ::Rcpp::internal::const_generic_proxy<VECSXP> Rcpp::traits::r_vector_const_proxy< VECSXP >::type

Definition at line 231 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.630 Rcpp::traits::r_vector_element_converter< RTYPE > Struct Template Reference

`#include <converter.h>`

Public Types

- typedef ::Rcpp::internal::element_converter< RTYPE > type

6.630.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_element_converter< RTYPE >

Definition at line 94 of file converter.h.

6.630.2 Member Typedef Documentation

6.630.2.1 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

6.631 Rcpp::traits::r_vector_element_converter< EXPRSXP > Struct Template Reference

`#include <converter.h>`

Public Types

- typedef ::Rcpp::internal::generic_element_converter< EXPRSXP > type

6.631.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_element_converter< EXPRSXP >

Definition at line 103 of file converter.h.

6.631.2 Member Typedef Documentation

6.631.2.1 typedef ::Rcpp::internal::generic_element_converter<EXPRSXP> Rcpp::traits::r_vector_element_converter< EXPRSXP >::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

6.632 Rcpp::traits::r_vector_element_converter< STRSXP > Struct Template Reference

`#include <converter.h>`

Public Types

- typedef ::Rcpp::internal::string_element_converter< STRSXP > type

6.632.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_element_converter< STRSXP >

Definition at line 97 of file converter.h.

6.632.2 Member Typedef Documentation

6.632.2.1 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

6.633 Rcpp::traits::r_vector_element_converter< VECSXP > Struct Template Reference

`#include <converter.h>`

Public Types

- typedef ::Rcpp::internal::generic_element_converter< VECSXP > type

6.633.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_element_converter< VECSXP >

Definition at line 100 of file converter.h.

6.633.2 Member Typedef Documentation

6.633.2.1 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

6.634 Rcpp::traits::r_vector_iterator< RTYPE > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef storage_type< RTYPE >::type * type

6.634.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_iterator< RTYPE >

Definition at line 45 of file 00_forward_proxy.h.

6.634.2 Member Typedef Documentation

6.634.2.1 template<int RTYPE> typedef storage_type<RTYPE>::type* Rcpp::traits::r_vector_iterator< RTYPE >::type

Definition at line 239 of file proxy.h.

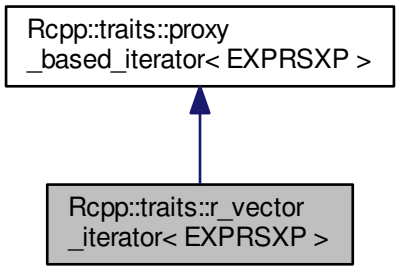
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

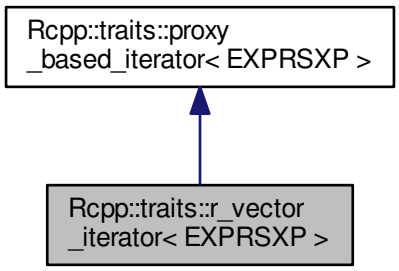
6.635 Rcpp::traits::r_vector_iterator< EXPRSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_iterator< EXPRSXP >:



Collaboration diagram for Rcpp::traits::r_vector_iterator< EXPRSXP >:



Additional Inherited Members

6.635.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_iterator< **EXPRSXP** >

Definition at line 250 of file proxy.h.

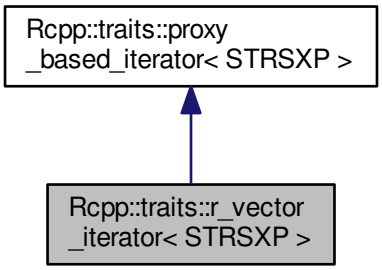
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

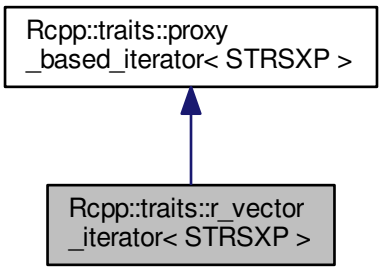
6.636 Rcpp::traits::r_vector_iterator< STRSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_iterator< STRSXP >:



Collaboration diagram for Rcpp::traits::r_vector_iterator< STRSXP >:



Additional Inherited Members

6.636.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_iterator< STRSXP >

Definition at line 251 of file proxy.h.

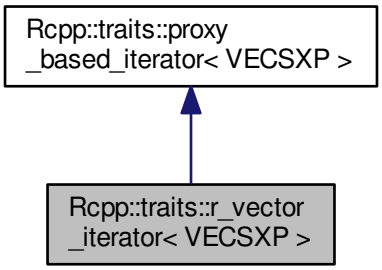
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

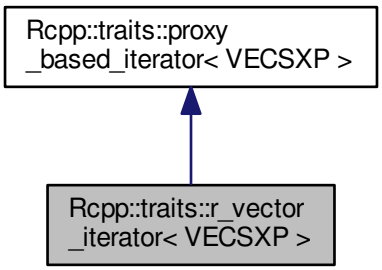
6.637 Rcpp::traits::r_vector_iterator< VECSXP > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_iterator< VECSXP >:



Collaboration diagram for Rcpp::traits::r_vector_iterator< VECSXP >:



Additional Inherited Members

6.637.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_iterator< VECSXP >

Definition at line 249 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_name_proxy< RTYPE > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef ::Rcpp::internal::simple_name_proxy< RTYPE > type

6.638.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_name_proxy< RTYPE >

Definition at line 41 of file 00_forward_proxy.h.

6.638.2 Member Typedef Documentation

6.638.2.1 template<int RTYPE> typedef ::Rcpp::internal::simple_name_proxy<RTYPE> Rcpp::traits::r_vector_name_proxy< RTYPE >::type

Definition at line 197 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

6.639 Rcpp::traits::r_vector_name_proxy< EXPRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::generic_name_proxy< EXPRSXP > type

6.639.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_name_proxy< EXPRSXP >

Definition at line 205 of file proxy.h.

6.639.2 Member Typedef Documentation

6.639.2.1 typedef ::Rcpp::internal::generic_name_proxy<EXPRSXP> Rcpp::traits::r_vector_name_proxy< EXPRSXP >::type

Definition at line 206 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_name_proxy< STRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::string_name_proxy< STRSXP > type

6.640.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_name_proxy< STRSXP >

Definition at line 199 of file proxy.h.

6.640.2 Member Typedef Documentation

6.640.2.1 typedef ::Rcpp::internal::string_name_proxy<STRSXP> Rcpp::traits::r_vector_name_proxy< STRSXP >::type

Definition at line 200 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.641 Rcpp::traits::r_vector_name_proxy< VECSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::generic_name_proxy< VECSXP > type

6.641.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_name_proxy< VECSXP >

Definition at line 202 of file proxy.h.

6.641.2 Member Typedef Documentation

6.641.2.1 typedef ::Rcpp::internal::generic_name_proxy<VECSXP> Rcpp::traits::r_vector_name_proxy< VECSXP >::type

Definition at line 203 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.642 Rcpp::traits::r_vector_proxy< RTYPE > Struct Template Reference

`#include <00_forward_proxy.h>`

Public Types

- typedef storage_type< RTYPE >::type & type

6.642.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_proxy< RTYPE >

Definition at line 42 of file 00_forward_proxy.h.

6.642.2 Member Typedef Documentation

6.642.2.1 template<int RTYPE> typedef storage_type<RTYPE>::type& Rcpp::traits::r_vector_proxy< RTYPE >::type

Definition at line 211 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

6.643 Rcpp::traits::r_vector_proxy< EXPRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::generic_proxy< EXPRSXP > type

6.643.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_proxy< EXPRSXP >

Definition at line 216 of file proxy.h.

6.643.2 Member Typedef Documentation

6.643.2.1 typedef ::Rcpp::internal::generic_proxy<EXPRSXP> Rcpp::traits::r_vector_proxy< EXPRSXP >::type

Definition at line 217 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.644 Rcpp::traits::r_vector_proxy< STRSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::string_proxy< STRSXP > type

6.644.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_proxy< STRSXP >

Definition at line 213 of file proxy.h.

6.644.2 Member Typedef Documentation

6.644.2.1 typedef ::Rcpp::internal::string_proxy<STRSXP> Rcpp::traits::r_vector_proxy< STRSXP >::type

Definition at line 214 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.645 Rcpp::traits::r_vector_proxy< VECSXP > Struct Template Reference

`#include <proxy.h>`

Public Types

- typedef ::Rcpp::internal::generic_proxy< VECSXP > type

6.645.1 Detailed Description

template<>
struct Rcpp::traits::r_vector_proxy< VECSXP >

Definition at line 219 of file proxy.h.

6.645.2 Member Typedef Documentation

6.645.2.1 typedef ::Rcpp::internal::generic_proxy<VECSXP> Rcpp::traits::r_vector_proxy< VECSXP >::type

Definition at line 220 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.646 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

6.646.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Range< RTYPE, NA, T >

Definition at line 29 of file range.h.

6.646.2 Member Typedef Documentation

6.646.2.1 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.

6.646.3 Constructor & Destructor Documentation

6.646.3.1 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.

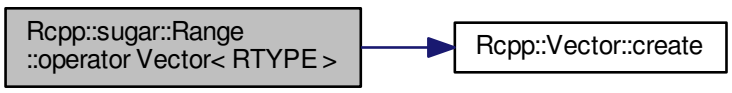
6.646.4 Member Function Documentation

6.646.4.1 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:



6.646.5 Member Data Documentation

6.646.5.1 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 >().

6.646.5.2 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 >().

6.646.5.3 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 >().

6.646.5.4 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 >().

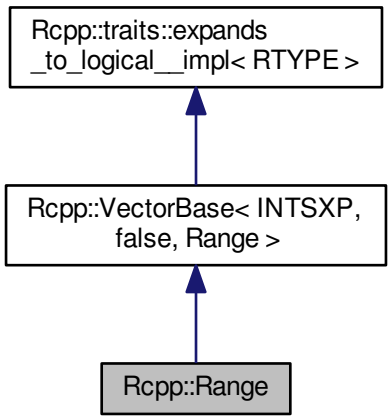
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/range.h

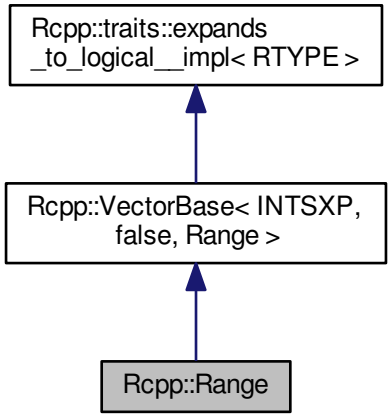
6.647 Rcpp::Range Class Reference

`#include <Range.h>`

Inheritance diagram for Rcpp::Range:



Collaboration diagram for Rcpp::Range:



Public Member Functions

- Range (int start_, int end__)
- int size () const
- int operator[ ] (int 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)
- int get_start () const
- int get_end () const

Private Attributes

- int start
- int end_

Additional Inherited Members

6.647.1 Detailed Description

Definition at line 27 of file Range.h.

6.647.2 Constructor & Destructor Documentation

6.647.2.1 Rcpp::Range::Range (int *start_*, int *end__*) [inline]

Definition at line 29 of file Range.h.

Referenced by operator+(), and operator-().

6.647.3 Member Function Documentation

6.647.3.1 int Rcpp::Range::get_end ( ) const [inline]

Definition at line 81 of file Range.h.

References end_.

6.647.3.2 int Rcpp::Range::get_start ( ) const [inline]

Definition at line 80 of file Range.h.

References start.

6.647.3.3 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:

6.647.3.4 Range& Rcpp::Range::operator++ ( ) [inline]

Definition at line 43 of file Range.h.

References end_, and start.

6.647.3.5 Range Rcpp::Range::operator++ (int) [inline]

Definition at line 47 of file Range.h.

6.647.3.6 Range& Rcpp::Range::operator+= (int *n*) [inline]

Definition at line 63 of file Range.h.

References end_, and start.

6.647.3.7 Range Rcpp::Range::operator- (int *n*) [inline]

Definition at line 76 of file Range.h.

References end_, Range(), and start.

Here is the call graph for this function:

6.647.3.8 Range& Rcpp::Range::operator-- ( ) [inline]

Definition at line 53 of file Range.h.

References end_, and start.

6.647.3.9 Range Rcpp::Range::operator-- (int) [inline]

Definition at line 57 of file Range.h.

6.647.3.10 Range& Rcpp::Range::operator-= (int *n*) [inline]

Definition at line 68 of file Range.h.

References end_, and start.

6.647.3.11 int Rcpp::Range::operator[ ] (int *i*) const [inline]

Definition at line 39 of file Range.h.

References start.

6.647.3.12 int Rcpp::Range::size ( ) const [inline]

Definition at line 35 of file Range.h.

References end_, and start.

6.647.4 Member Data Documentation

6.647.4.1 int Rcpp::Range::end_ [private]

Definition at line 85 of file Range.h.

Referenced by get_end(), operator+(), operator++(), operator+=(), operator-(), operator--(), operator-=(), and size().

6.647.4.2 int Rcpp::Range::start [private]

Definition at line 84 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

6.648 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

6.648.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Range< RTYPE, false, T >

Definition at line 58 of file range.h.

6.648.2 Member Typedef Documentation

6.648.2.1 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.

6.648.3 Constructor & Destructor Documentation

6.648.3.1 template<int RTYPE, typename T > Rcpp::sugar::Range< RTYPE, false, T >::Range (const T & *obj_*) [inline]

Definition at line 62 of file range.h.

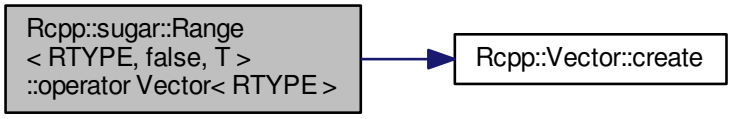
6.648.4 Member Function Documentation

6.648.4.1 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:



6.648.5 Member Data Documentation

6.648.5.1 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Range< RTYPE, false, T >::current [private]

Definition at line 79 of file range.h.

6.648.5.2 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Range< RTYPE, false, T >::max_ [private]

Definition at line 79 of file range.h.

6.648.5.3 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Range< RTYPE, false, T >::min_ [private]

Definition at line 79 of file range.h.

6.648.5.4 template<int RTYPE, typename T > const T& Rcpp::sugar::Range< RTYPE, false, T >::obj [private]

Definition at line 78 of file range.h.

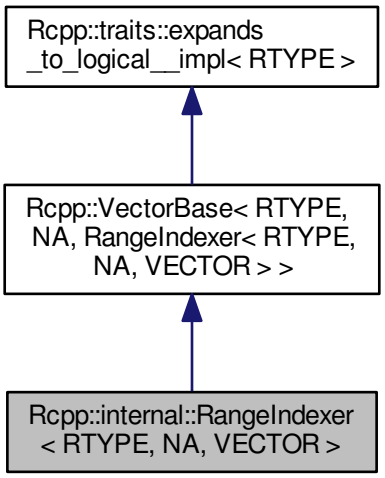
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/range.h

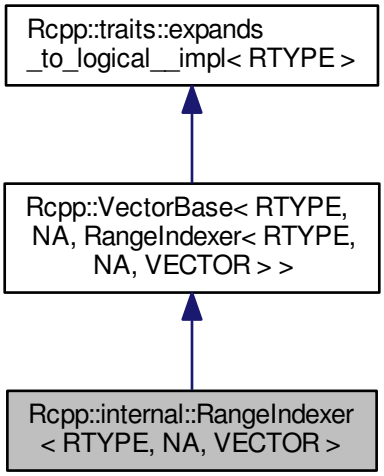
6.649 Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR > Class Template Reference

`#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*= (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_

6.649.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
class Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >

Definition at line 53 of file RangeIndexer.h.

6.649.2 Member Typedef Documentation

6.649.2.1 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.

6.649.2.2 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.

6.649.3 Constructor & Destructor Documentation

6.649.3.1 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.

6.649.4 Member Function Documentation

6.649.4.1 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.

6.649.4.2 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.

6.649.4.3 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.

6.649.4.4 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.

6.649.4.5 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.

6.649.4.6 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.

6.649.4.7 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_.

6.649.5 Member Data Documentation

6.649.5.1 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().

6.649.5.2 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

6.650 Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression Struct Reference

`#include <VectorBase.h>`

6.650.1 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

6.651 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

6.651.1 Detailed Description

template<typename T>
class Rcpp::ReferenceInputParameter< T >

Definition at line 43 of file InputParameter.h.

6.651.2 Member Typedef Documentation

6.651.2.1 template<typename T > typedef T& Rcpp::ReferenceInputParameter< T >::reference

Definition at line 45 of file InputParameter.h.

6.651.3 Constructor & Destructor Documentation

6.651.3.1 template<typename T > Rcpp::ReferenceInputParameter< T >::ReferenceInputParameter (SEXP *x_*) [inline]

Definition at line 46 of file InputParameter.h.

6.651.4 Member Function Documentation

6.651.4.1 template<typename T > Rcpp::ReferenceInputParameter< T >::operator reference ( ) [inline]

Definition at line 48 of file InputParameter.h.

6.651.5 Member Data Documentation

6.651.5.1 template<typename T > T Rcpp::ReferenceInputParameter< T >::obj [private]

Definition at line 51 of file InputParameter.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.652 Rcpp::traits::remove_const< _Tp > Struct Template Reference

const-volatile modifications [4.7.1].

`#include <remove_const.h>`

Public Types

- typedef _Tp type

6.652.1 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.

6.652.2 Member Typedef Documentation

6.652.2.1 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

6.653 Rcpp::traits::remove_const< _Tp const > Struct Template Reference

`#include <remove_const.h>`

Public Types

- typedef _Tp type

6.653.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_const< _Tp const >

Definition at line 34 of file remove_const.h.

6.653.2 Member Typedef Documentation

6.653.2.1 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

6.654 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

6.654.1 Detailed Description

template<typename T>
struct Rcpp::traits::remove_const_and_reference< T >

Definition at line 29 of file remove_const_and_reference.h.

6.654.2 Member Typedef Documentation

6.654.2.1 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

6.655 Rcpp::traits::remove_reference< _Tp > Struct Template Reference

reference modifications [4.7.2].

`#include <remove_reference.h>`

Public Types

- typedef _Tp type

6.655.1 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.

6.655.2 Member Typedef Documentation

6.655.2.1 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

6.656 Rcpp::traits::remove_reference< _Tp & > Struct Template Reference

`#include <remove_reference.h>`

Public Types

- typedef _Tp type

6.656.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_reference< _Tp & >

Definition at line 34 of file remove_reference.h.

6.656.2 Member Typedef Documentation

6.656.2.1 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

6.657 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

6.657.1 Detailed Description

template<typename SET>
class Rcpp::sugar::RemoveFromSet< SET >

Definition at line 29 of file setdiff.h.

6.657.2 Constructor & Destructor Documentation

6.657.2.1 template<typename SET > Rcpp::sugar::RemoveFromSet< SET >::RemoveFromSet (SET & *set_*) [inline]

Definition at line 31 of file setdiff.h.

6.657.3 Member Function Documentation

6.657.3.1 template<typename SET > template<typename T > void Rcpp::sugar::RemoveFromSet< SET >::operator() (T *value*) [inline]

Definition at line 34 of file setdiff.h.

6.657.4 Member Data Documentation

6.657.4.1 template<typename SET > SET& Rcpp::sugar::RemoveFromSet< SET >::set [private]

Definition at line 39 of file setdiff.h.

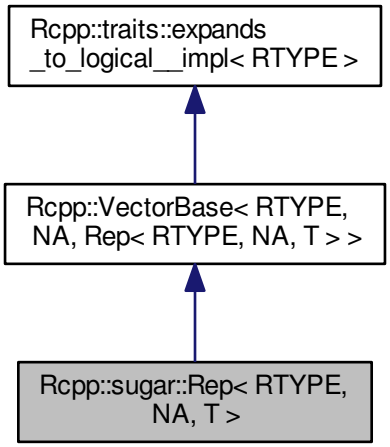
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

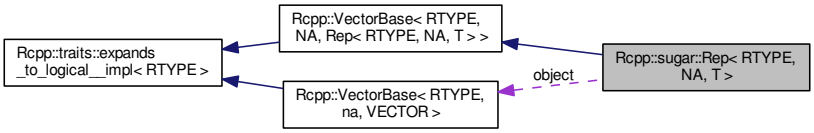
6.658 Rcpp::sugar::Rep< RTYPE, NA, T > Class Template Reference

`#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

6.658.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep< RTYPE, NA, T >

Definition at line 29 of file rep.h.

6.658.2 Member Typedef Documentation

6.658.2.1 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.

6.658.2.2 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.

6.658.3 Constructor & Destructor Documentation

6.658.3.1 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.

6.658.4 Member Function Documentation

6.658.4.1 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.

6.658.4.2 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.

6.658.5 Member Data Documentation

6.658.5.1 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[ ](), Rcpp::rep(), Rcpp::sugar::Rep< RTYPE, NA, T >::size(), and Rcpp::sugar::Rep_Single< T >::size().

6.658.5.2 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.

6.658.5.3 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().

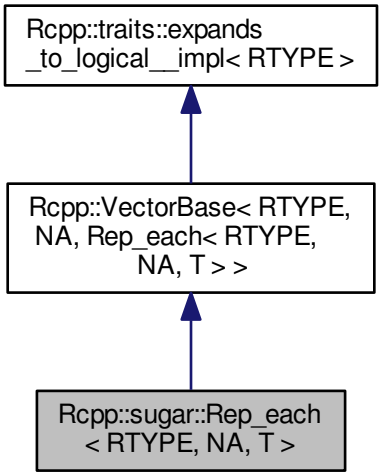
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep.h

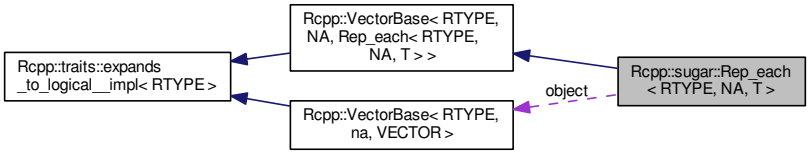
6.659 Rcpp::sugar::Rep_each< RTYPE, NA, T > Class Template Reference

`#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

6.659.1 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.

6.659.2 Member Typedef Documentation

6.659.2.1 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.

6.659.2.2 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.

6.659.3 Constructor & Destructor Documentation

6.659.3.1 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.

6.659.4 Member Function Documentation

6.659.4.1 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.

6.659.4.2 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.

6.659.5 Member Data Documentation

6.659.5.1 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().

6.659.5.2 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.

6.659.5.3 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[ ](), Rcpp::rep_each(), and Rcpp::sugar::Rep_↩each< RTYPE, NA, T >::size().

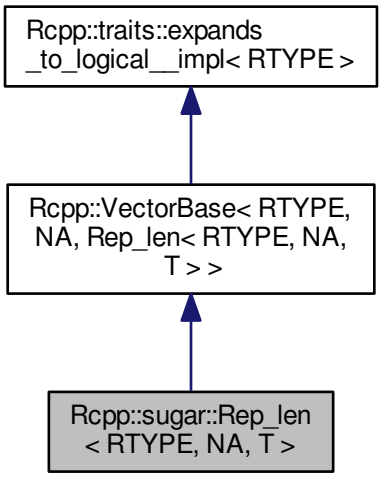
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep_each.h

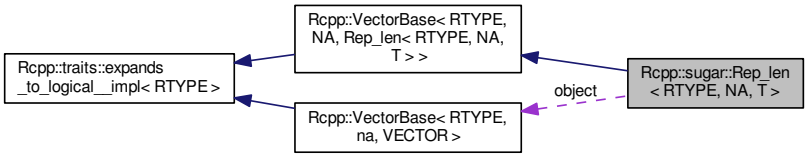
6.660 Rcpp::sugar::Rep_len< RTYPE, NA, T > Class Template Reference

`#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

6.660.1 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.

6.660.2 Member Typedef Documentation

6.660.2.1 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.

6.660.2.2 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.

6.660.3 Constructor & Destructor Documentation

6.660.3.1 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.

6.660.4 Member Function Documentation

6.660.4.1 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.

6.660.4.2 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.

6.660.5 Member Data Documentation

6.660.5.1 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::rep_len(), and Rcpp::sugar::Rep_len< RTYPE, NA, T >::size().

6.660.5.2 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[]().

6.660.5.3 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.

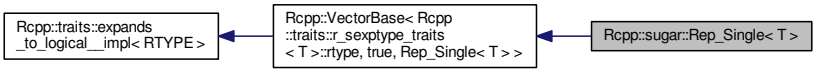
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep_len.h

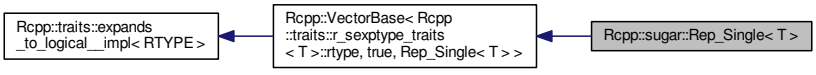
6.661 Rcpp::sugar::Rep_Single< T > Class Template Reference

`#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 i) const
- R_xlen_t size () const

Private Attributes

- const T & x
- R_xlen_t n

Additional Inherited Members

6.661.1 Detailed Description

template<typename T>
class Rcpp::sugar::Rep_Single< T >

Definition at line 48 of file rep.h.

6.661.2 Constructor & Destructor Documentation

6.661.2.1 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.

6.661.3 Member Function Documentation

6.661.3.1 template<typename T> T Rcpp::sugar::Rep_Single< T >::operator[ ] (R_xlen_t *i*) const [inline]

Definition at line 56 of file rep.h.

6.661.3.2 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< RTYPE, NA, T >::n.

6.661.4 Member Data Documentation

6.661.4.1 template<typename T> R_xlen_t Rcpp::sugar::Rep_Single< T >::n [private]

Definition at line 63 of file rep.h.

6.661.4.2 template<typename T> const T& Rcpp::sugar::Rep_Single< T >::x [private]

Definition at line 62 of file rep.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep.h

6.662 Rcpp::result< T > Class Template Reference

`#include <Module.h>`

Public Member Functions

- result (T *ptr_)
- operator T * ()

Private Attributes

- T * ptr

6.662.1 Detailed Description

template<typename T>
class Rcpp::result< T >

Definition at line 56 of file Module.h.

6.662.2 Constructor & Destructor Documentation

6.662.2.1 template<typename T > Rcpp::result< T >::result (T * *ptr_*) [inline]

Definition at line 58 of file Module.h.

6.662.3 Member Function Documentation

6.662.3.1 template<typename T > Rcpp::result< T >::operator T *( ) [inline]

Definition at line 59 of file Module.h.

6.662.4 Member Data Documentation

6.662.4.1 template<typename T > T* Rcpp::result< T >::ptr [private]

Definition at line 61 of file Module.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.663 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

6.663.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::median_detail::result< RTYPE >

Definition at line 35 of file median.h.

6.663.2 Member Typedef Documentation

6.663.2.1 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.

6.663.3 Member Enumeration Documentation

6.663.3.1 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

6.664 Rcpp::sugar::median_detail::result< INTSXP > Struct Template Reference

`#include <median.h>`

Public Types

- enum { rtype = REALSXP }
- typedef double type

6.664.1 Detailed Description

template<>
struct Rcpp::sugar::median_detail::result< INTSXP >

Definition at line 41 of file median.h.

6.664.2 Member Typedef Documentation

6.664.2.1 typedef double Rcpp::sugar::median_detail::result< INTSXP >::type

Definition at line 42 of file median.h.

6.664.3 Member Enumeration Documentation

6.664.3.1 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

6.665 Rcpp::sugar::median_detail::result< STRSXP > Struct Template Reference

`#include <median.h>`

Public Types

- enum { rtype = STRSXP }
- typedef Rcpp::String type

6.665.1 Detailed Description

template<>
struct Rcpp::sugar::median_detail::result< STRSXP >

Definition at line 47 of file median.h.

6.665.2 Member Typedef Documentation

6.665.2.1 typedef Rcpp::String Rcpp::sugar::median_detail::result< STRSXP >::type

Definition at line 48 of file median.h.

6.665.3 Member Enumeration Documentation

6.665.3.1 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

6.666 Rcpp::traits::result_of< T > Struct Template Reference

`#include <result_of.h>`

Public Types

- typedef T::result_type type

6.666.1 Detailed Description

template<typename T>
struct Rcpp::traits::result_of< T >

Definition at line 30 of file result_of.h.

6.666.2 Member Typedef Documentation

6.666.2.1 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

6.667 Rcpp::traits::result_of< RESULT_TYPE(*)(INPUT_TYPE) > Struct Template Reference

`#include <result_of.h>`

Public Types

- typedef RESULT_TYPE type

6.667.1 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.

6.667.2 Member Typedef Documentation

6.667.2.1 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

6.668 Rcpp::traits::result_of< RESULT_TYPE(*)(U1, U2) > Struct Template Reference

`#include <result_of.h>`

Public Types

- typedef RESULT_TYPE type

6.668.1 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.

6.668.2 Member Typedef Documentation

6.668.2.1 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

6.669 Rcpp::traits::result_of< RESULT_TYPE(*)(U1, U2, U3) > Struct Template Reference

`#include <result_of.h>`

Public Types

- typedef RESULT_TYPE type

6.669.1 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.

6.669.2 Member Typedef Documentation

6.669.2.1 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.

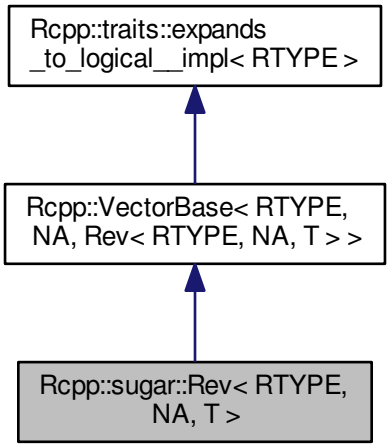
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result_of.h

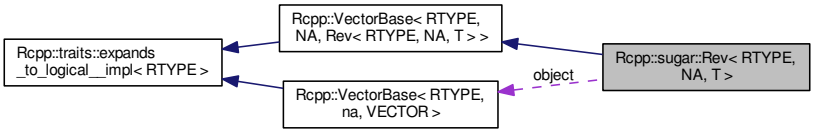
6.670 Rcpp::sugar::Rev< RTYPE, NA, T > Class Template Reference

`#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

6.670.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rev< RTYPE, NA, T >

Definition at line 29 of file rev.h.

6.670.2 Member Typedef Documentation

6.670.2.1 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.

6.670.2.2 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.

6.670.3 Constructor & Destructor Documentation

6.670.3.1 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.

6.670.4 Member Function Documentation

6.670.4.1 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.

6.670.4.2 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.

6.670.5 Member Data Documentation

6.670.5.1 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().

6.670.5.2 template<int RTYPE, bool NA, typename T > const VEC_TYPE& Rcpp::sugar::Rev< RTYPE, NA, T >::object [private]

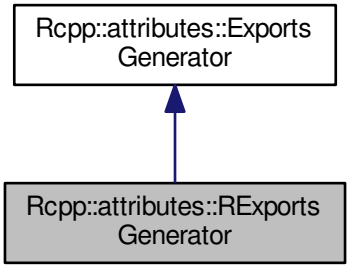
Definition at line 43 of file rev.h.

The documentation for this class was generated from the following file:

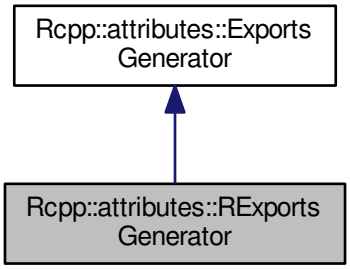
- inst/include/Rcpp/sugar/functions/rev.h

6.671 Rcpp::attributes::RExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::RExportsGenerator:



Collaboration diagram for Rcpp::attributes::RExportsGenerator:



Public Member Functions

- RExportsGenerator (const std::string &packageDir, const std::string &package, const std::string &fileSep)
- virtual void writeBegin ()
- virtual void writeEnd ()
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)

Additional Inherited Members

6.671.1 Detailed Description

Definition at line 709 of file attributes.cpp.

6.671.2 Constructor & Destructor Documentation

6.671.2.1 Rcpp::attributes::RExportsGenerator::RExportsGenerator (const std::string & *packageDir,* const std::string & *package,* const std::string & *fileSep*)

Definition at line 2203 of file attributes.cpp.

6.671.3 Member Function Documentation

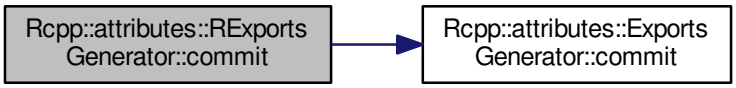
6.671.3.1 bool Rcpp::attributes::RExportsGenerator::commit (const std::vector< std::string > & *includes*) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2285 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit().

Here is the call graph for this function:



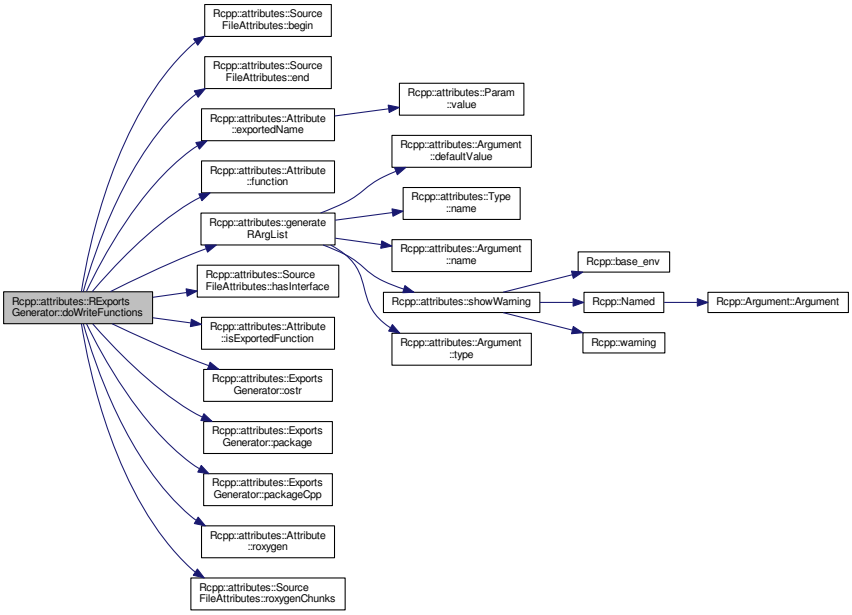
6.671.3.2 void Rcpp::attributes::RExportsGenerator::doWriteFunctions (const SourceFileAttributes & *attributes,* bool *verbose*) [private],[virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2213 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::Attribute::exportedName(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::generateRArgList(), Rcpp::attributes::SourceFileAttributes::hasInterface(), Rcpp::attributes::Attribute::isExportedFunction(), Rcpp::attributes::kInterfaceR, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), Rcpp::attributes::ExportsGenerator::packageCpp(), Rcpp::attributes::Attribute::roxygen(), and Rcpp::attributes::SourceFileAttributes::roxygenChunks().

Here is the call graph for this function:



6.671.3.3 virtual void Rcpp::attributes::RExportsGenerator::writeBegin ( ) [inline], [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 715 of file attributes.cpp.

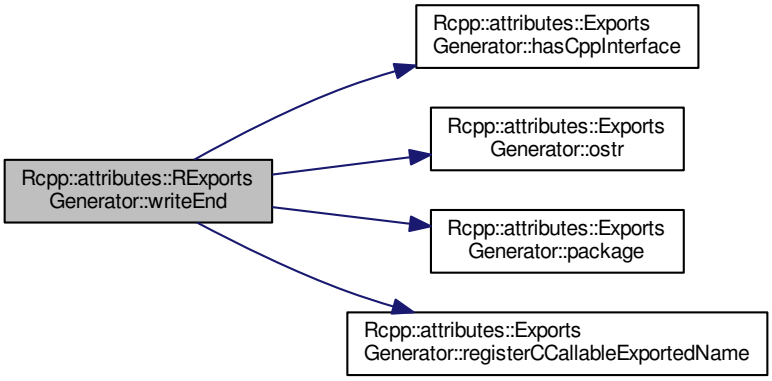
6.671.3.4 void Rcpp::attributes::RExportsGenerator::writeEnd ( ) [virtual]

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2273 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:



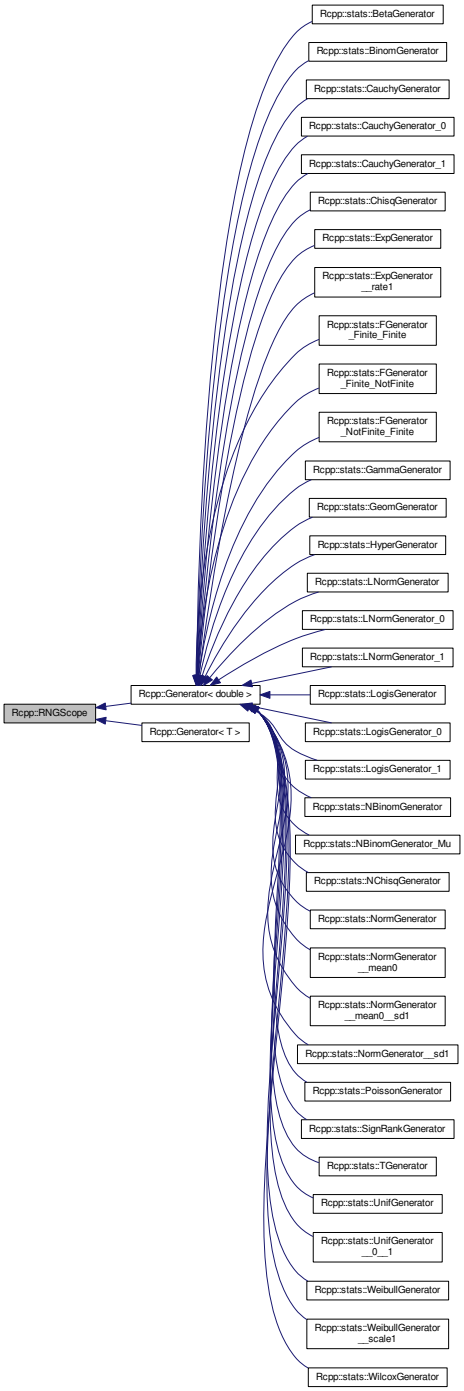
The documentation for this class was generated from the following file:

- src/attributes.cpp

6.672 Rcpp::RNGScope Class Reference

```
#include <random.h>
```

Inheritance diagram for Rcpp::RNGScope:



Public Member Functions

- RNGScope ()
- ~RNGScope ()

6.672.1 Detailed Description

Definition at line 27 of file random.h.

6.672.2 Constructor & Destructor Documentation

6.672.2.1 Rcpp::RNGScope::RNGScope ( ) [inline]

Definition at line 29 of file random.h.

References Rcpp::internal::enterRNGScope().

Here is the call graph for this function:

6.672.2.2 Rcpp::RNGScope::~RNGScope ( ) [inline]

Definition at line 30 of file random.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/stats/random/random.h

6.673 Rcpp::RObjectMethods< Class > Class Template Reference

`#include <RObjectMethods.h>`

Public Member Functions

- bool isNULL () const
- int sexp_type () const
- bool isObject () const
- bool isS4 () const

6.673.1 Detailed Description

template<typename Class>
class Rcpp::RObjectMethods< Class >

Definition at line 24 of file RObjectMethods.h.

6.673.2 Member Function Documentation

6.673.2.1 template<typename Class> bool Rcpp::RObjectMethods< Class >::isNULL ( ) const [inline]

Definition at line 27 of file RObjectMethods.h.

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::from_list().

6.673.2.2 template<typename Class> bool Rcpp::RObjectMethods< Class >::isObject ( ) const [inline]

Definition at line 35 of file RObjectMethods.h.

6.673.2.3 template<typename Class> bool Rcpp::RObjectMethods< Class >::isS4 ( ) const [inline]

Definition at line 39 of file RObjectMethods.h.

6.673.2.4 template<typename Class> int Rcpp::RObjectMethods< Class >::sexp_type ( ) const [inline]

Definition at line 31 of file RObjectMethods.h.

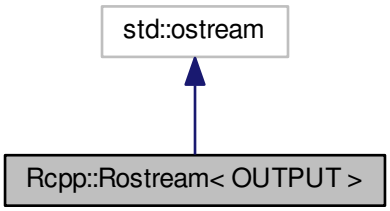
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/RObjectMethods.h

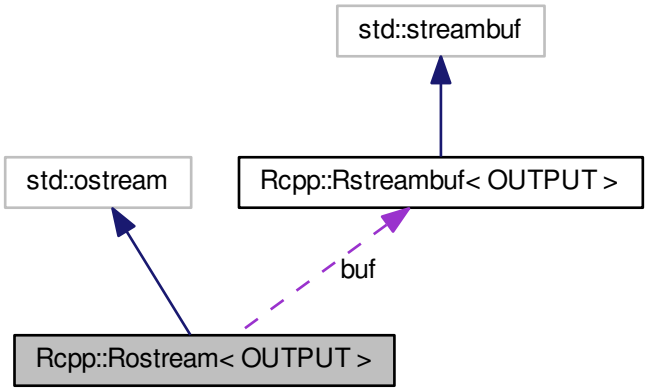
6.674 Rcpp::Rostream< OUTPUT > Class Template Reference

`#include <Rstreambuf.h>`

Inheritance diagram for Rcpp::Rostream< OUTPUT >:



Collaboration diagram for Rcpp::Rostream< OUTPUT >:



Public Member Functions

- Rostream ()
- ~Rostream ()

Private Types

- typedef Rstreambuf< OUTPUT > Buffer

Private Attributes

- Buffer * buf

6.674.1 Detailed Description

template<bool OUTPUT>
class Rcpp::Rostream< OUTPUT >

Definition at line 44 of file Rstreambuf.h.

6.674.2 Member Typedef Documentation

6.674.2.1 template<bool OUTPUT> typedef Rstreambuf<OUTPUT> Rcpp::Rostream< OUTPUT >::Buffer [private]

Definition at line 45 of file Rstreambuf.h.

6.674.3 Constructor & Destructor Documentation

6.674.3.1 template<bool OUTPUT> Rcpp::Rostream< OUTPUT >::Rostream ( ) [inline]

Definition at line 48 of file Rstreambuf.h.

6.674.3.2 template<bool OUTPUT> Rcpp::Rostream< OUTPUT >::~Rostream ( ) [inline]

Definition at line 52 of file Rstreambuf.h.

6.674.4 Member Data Documentation

6.674.4.1 template<bool OUTPUT> Buffer* Rcpp::Rostream< OUTPUT >::buf [private]

Definition at line 46 of file Rstreambuf.h.

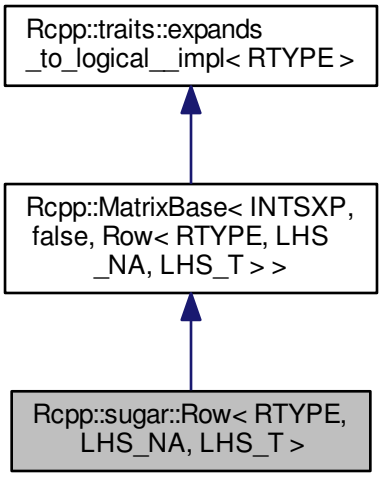
The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/Rstreambuf.h

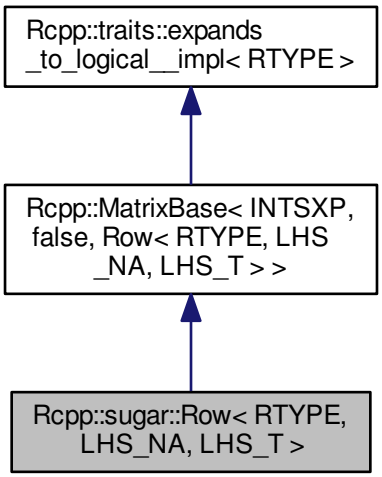
6.675 Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T > Class Template Reference

`#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)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Attributes

- int nr
- int nc

6.675.1 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.

6.675.2 Member Typedef Documentation

6.675.2.1 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.

6.675.3 Constructor & Destructor Documentation

6.675.3.1 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.

6.675.4 Member Function Documentation

6.675.4.1 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.

6.675.4.2 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.

6.675.4.3 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.

6.675.4.4 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.

6.675.5 Member Data Documentation

6.675.5.1 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().

6.675.5.2 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().

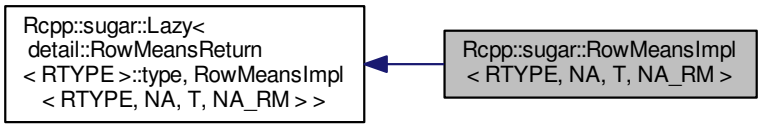
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/row.h

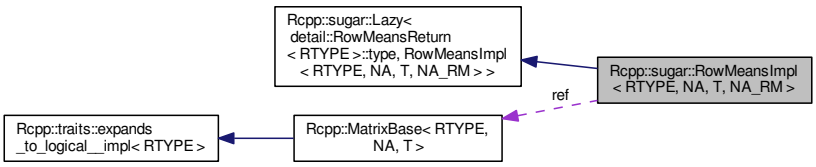
6.676 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference

`#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

6.676.1 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.

6.676.2 Member Typedef Documentation

6.676.2.1 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.

6.676.2.2 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.

6.676.2.3 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.

6.676.3 Constructor & Destructor Documentation

6.676.3.1 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.

6.676.4 Member Function Documentation

6.676.4.1 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.

6.676.5 Member Data Documentation

6.676.5.1 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.

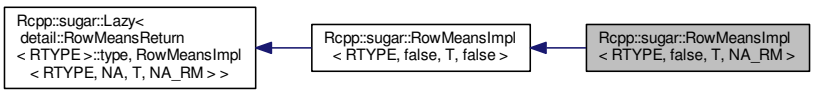
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

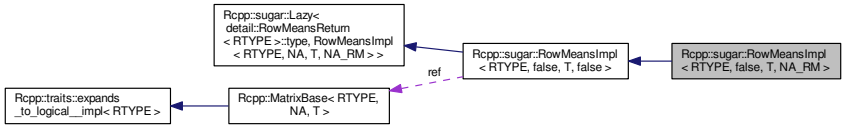
6.677 Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference

`#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

6.677.1 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.

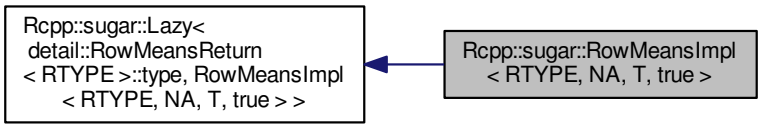
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

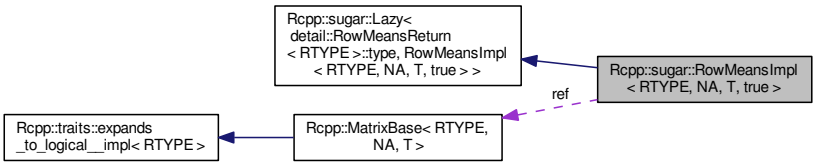
6.678 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true > Class Template Reference

`#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

6.678.1 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.

6.678.2 Member Typedef Documentation

6.678.2.1 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.

6.678.2.2 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.

6.678.2.3 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.

6.678.3 Constructor & Destructor Documentation

6.678.3.1 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.

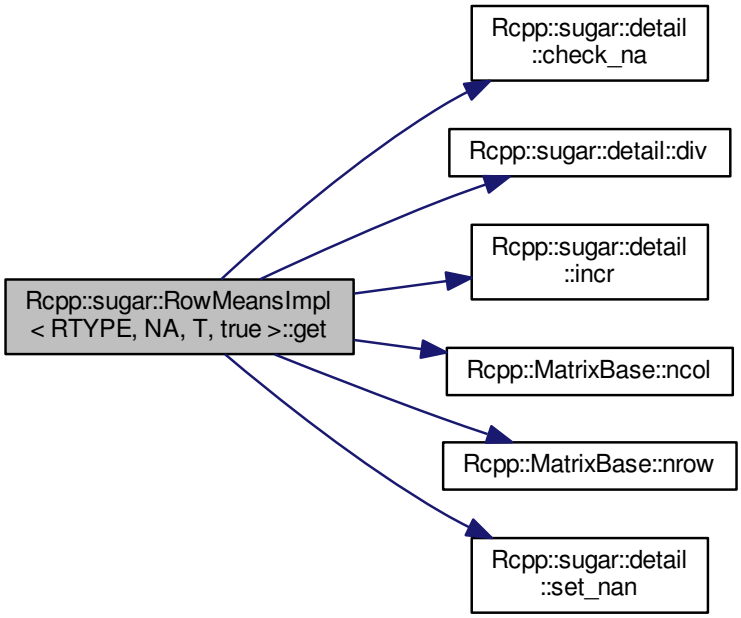
6.678.4 Member Function Documentation

6.678.4.1 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(), and Rcpp::sugar::detail::set_nan().

Here is the call graph for this function:



6.678.5 Member Data Documentation

6.678.5.1 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.

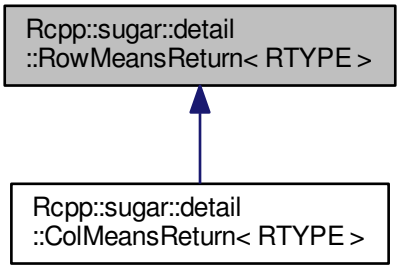
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.679 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

6.679.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::RowMeansReturn< RTYPE >

Definition at line 103 of file rowSums.h.

6.679.2 Member Typedef Documentation

6.679.2.1 template<int RTYPE> typedef Vector<REALSXP> Rcpp::sugar::detail::RowMeansReturn< RTYPE >::type

Definition at line 104 of file rowSums.h.

6.679.3 Member Enumeration Documentation

6.679.3.1 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

6.680 Rcpp::sugar::detail::RowMeansReturn< CPLXSXP > Struct Template Reference

`#include <rowSums.h>`

Public Types

- enum { rtype = CPLXSXP }
- typedef Vector< CPLXSXP > type

6.680.1 Detailed Description

template<>
struct Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >

Definition at line 109 of file rowSums.h.

6.680.2 Member Typedef Documentation

6.680.2.1 typedef Vector<CPLXSXP> Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >::type

Definition at line 110 of file rowSums.h.

6.680.3 Member Enumeration Documentation

6.680.3.1 anonymous enum

Enumerator

rtype

Definition at line 111 of file rowSums.h.

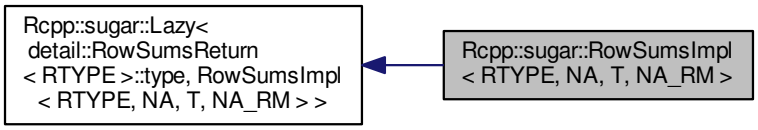
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

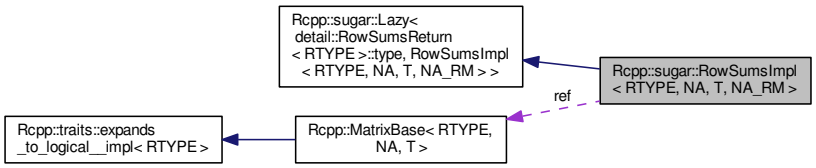
6.681 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference

`#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

6.681.1 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.

6.681.2 Member Typedef Documentation

6.681.2.1 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.

6.681.2.2 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.

6.681.2.3 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.

6.681.3 Constructor & Destructor Documentation

6.681.3.1 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.

6.681.4 Member Function Documentation

6.681.4.1 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.

6.681.5 Member Data Documentation

6.681.5.1 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.

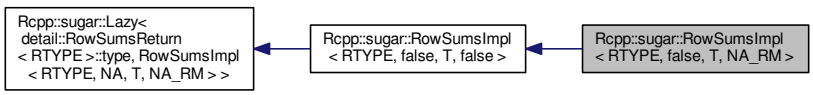
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

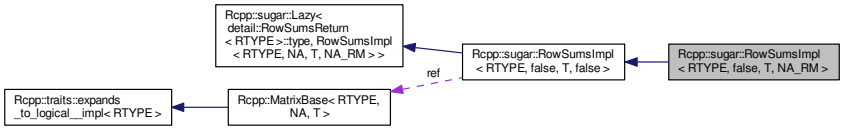
6.682 Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference

`#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

6.682.1 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.

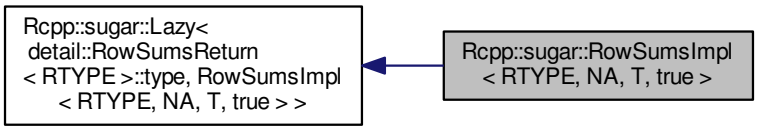
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

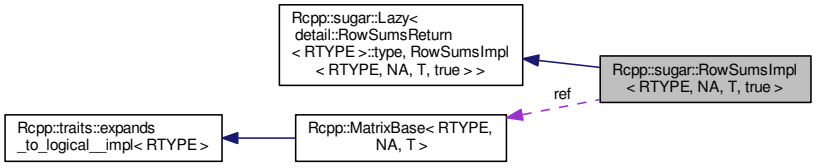
6.683 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true > Class Template Reference

`#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

6.683.1 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.

6.683.2 Member Typedef Documentation

6.683.2.1 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.

6.683.2.2 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.

6.683.2.3 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.

6.683.3 Constructor & Destructor Documentation

6.683.3.1 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.

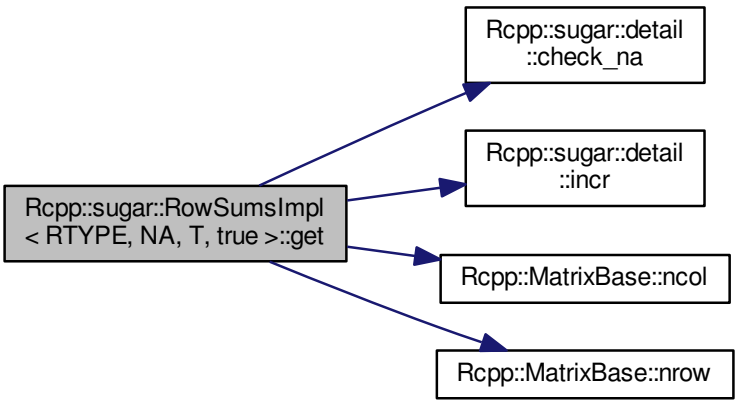
6.683.4 Member Function Documentation

6.683.4.1 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(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow().

Here is the call graph for this function:



6.683.5 Member Data Documentation

6.683.5.1 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.

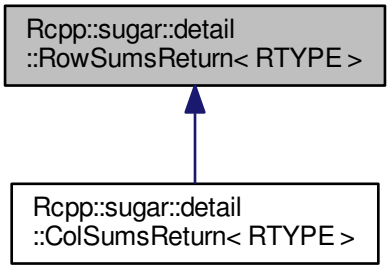
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.684 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

6.684.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::RowSumsReturn< RTYPE >

Definition at line 86 of file rowSums.h.

6.684.2 Member Typedef Documentation

6.684.2.1 template<int RTYPE> typedef Vector<RTYPE> Rcpp::sugar::detail::RowSumsReturn< RTYPE >::type

Definition at line 87 of file rowSums.h.

6.684.3 Member Enumeration Documentation

6.684.3.1 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

6.685 Rcpp::sugar::detail::RowSumsReturn< LGLSXP > Struct Template Reference

`#include <rowSums.h>`

Public Types

- enum { rtype = INTSXP }
- typedef Vector< INTSXP > type

6.685.1 Detailed Description

template<>
struct Rcpp::sugar::detail::RowSumsReturn< LGLSXP >

Definition at line 92 of file rowSums.h.

6.685.2 Member Typedef Documentation

6.685.2.1 typedef Vector<INTSXP> Rcpp::sugar::detail::RowSumsReturn< LGLSXP >::type

Definition at line 93 of file rowSums.h.

6.685.3 Member Enumeration Documentation

6.685.3.1 anonymous enum

Enumerator

rtype

Definition at line 94 of file rowSums.h.

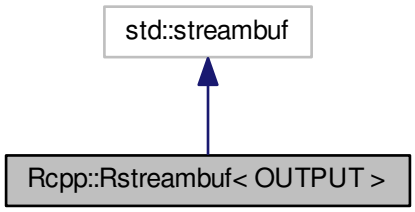
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

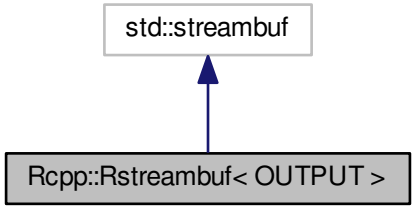
6.686 Rcpp::Rstreambuf< OUTPUT > Class Template Reference

`#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 xsputn (const char *s, std::streamsize n)
- virtual int overflow (int c=EOF)
- virtual int sync ()
- template<>
  std::streamsize xsputn (const char *s, std::streamsize num)
- template<>
  std::streamsize xsputn (const char *s, std::streamsize num)
- template<>
  int overflow (int c)
- template<>
  int overflow (int c)
- template<>
  int sync ()
- template<>
  int sync ()

6.686.1 Detailed Description

template<bool OUTPUT>
class Rcpp::Rstreambuf< OUTPUT >

Definition at line 31 of file Rstreambuf.h.

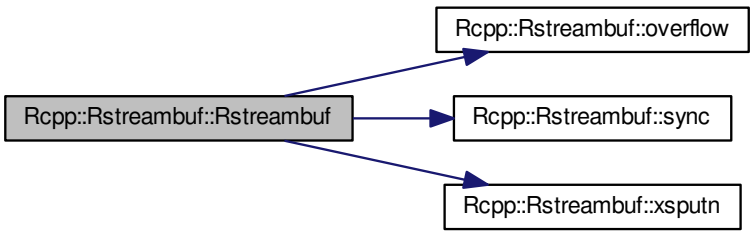
6.686.2 Constructor & Destructor Documentation

6.686.2.1 template<bool OUTPUT> Rcpp::Rstreambuf< OUTPUT >::Rstreambuf ( ) [inline]

Definition at line 33 of file Rstreambuf.h.

References Rcpp::Rstreambuf< OUTPUT >::overflow(), Rcpp::Rstreambuf< OUTPUT >::sync(), and Rcpp::Rstreambuf< OUTPUT >::xsputn().

Here is the call graph for this function:



6.686.3 Member Function Documentation

6.686.3.1 template<bool OUTPUT> virtual int Rcpp::Rstreambuf< OUTPUT >::overflow (int *c* = EOF) [protected], [virtual]

Referenced by Rcpp::Rstreambuf< OUTPUT >::Rstreambuf().

6.686.3.2 template<> int Rcpp::Rstreambuf< true >::overflow (int *c*) [inline], [protected]

Definition at line 69 of file Rstreambuf.h.

6.686.3.3 template<> int Rcpp::Rstreambuf< false >::overflow (int *c*) [inline], [protected]

Definition at line 73 of file Rstreambuf.h.

6.686.3.4 template<bool OUTPUT> virtual int Rcpp::Rstreambuf< OUTPUT >::sync ( ) [protected], [virtual]

Referenced by Rcpp::Rstreambuf< OUTPUT >::Rstreambuf().

6.686.3.5 template<> int Rcpp::Rstreambuf< true >::sync ( ) [inline], [protected]

Definition at line 78 of file Rstreambuf.h.

6.686.3.6 template<> int Rcpp::Rstreambuf< false >::sync ( ) [inline], [protected]

Definition at line 82 of file Rstreambuf.h.

6.686.3.7 template<bool OUTPUT> virtual std::streamsize Rcpp::Rstreambuf< OUTPUT >::xsputn (const char * *s*, std::streamsize *n*) [protected], [virtual]

Referenced by Rcpp::Rstreambuf< OUTPUT >::Rstreambuf().

6.686.3.8 template<> std::streamsize Rcpp::Rstreambuf< true >::xsputn (const char * *s*, std::streamsize *num*) [inline], [protected]

Definition at line 60 of file Rstreambuf.h.

6.686.3.9 template<> std::streamsize Rcpp::Rstreambuf< false >::xsputn (const char * *s*, std::streamsize *num*) [inline], [protected]

Definition at line 64 of file Rstreambuf.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/Rstreambuf.h

6.687 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 T ZERO ()
- static RCPP_CONSTEXPR T ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR int RTYPE = helper_type::RTYPE

6.687.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::rtype< T >

Definition at line 167 of file algorithm.h.

6.687.2 Member Typedef Documentation

6.687.2.1 template<typename T > typedef rtype_helper< typename ctype< T >::type > Rcpp::algorithm::helpers::rtype< T >::helper_type

Definition at line 169 of file algorithm.h.

6.687.2.2 template<typename T > typedef rtype_helper< typename ctype< T >::type >::type Rcpp::algorithm::helpers::rtype< T >::type

Definition at line 168 of file algorithm.h.

6.687.3 Member Function Documentation

6.687.3.1 template<typename T > static T Rcpp::algorithm::helpers::rtype< T >::NA ( ) [inline], [static]

Definition at line 171 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::algorithm::max(), Rcpp::algorithm::mean(), Rcpp::algorithm::min(), Rcpp::algorithm::helpers::log::operator()(), Rcpp::algorithm::helpers::exp::operator()(), Rcpp::algorithm::helpers::sqrt::operator()(), Rcpp::algorithm::prod(), and Rcpp::algorithm::sum().

6.687.3.2 template<typename T > static RCPP_CONSTEXPR T Rcpp::algorithm::helpers::rtype< T >::ONE ( ) [inline], [static]

Definition at line 173 of file algorithm.h.

Referenced by Rcpp::algorithm::max(), Rcpp::algorithm::max_nona(), Rcpp::algorithm::prod(), and Rcpp::algorithm::prod_nona().

6.687.3.3 template<typename T > static RCPP_CONSTEXPR T Rcpp::algorithm::helpers::rtype< T >::ZERO ( ) [inline], [static]

Definition at line 172 of file algorithm.h.

Referenced by Rcpp::algorithm::sum(), and Rcpp::algorithm::sum_nona().

6.687.4 Member Data Documentation

6.687.4.1 template<typename T > RCPP_CONSTEXPR int Rcpp::algorithm::helpers::rtype< T >::RTYPE = helper_type::RTYPE [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

6.688 Rcpp::algorithm::helpers::rtype_helper< T > Struct Template Reference

`#include <algorithm.h>`

6.688.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::rtype_helper< T >

Definition at line 138 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.689 Rcpp::algorithm::helpers::rtype_helper< double > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef double type

Static Public Member Functions

- static double NA ()
- static RCPP_CONSTEXPR double ZERO ()
- static RCPP_CONSTEXPR double ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR int RTYPE = REALSXP

6.689.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::rtype_helper< double >

Definition at line 149 of file algorithm.h.

6.689.2 Member Typedef Documentation

6.689.2.1 typedef double Rcpp::algorithm::helpers::rtype_helper< double >::type

Definition at line 150 of file algorithm.h.

6.689.3 Member Function Documentation

6.689.3.1 static double Rcpp::algorithm::helpers::rtype_helper< double >::NA ( ) [inline], [static]

Definition at line 152 of file algorithm.h.

6.689.3.2 static RCPP_CONSTEXPR double Rcpp::algorithm::helpers::rtype_helper< double >::ONE ( ) [inline], [static]

Definition at line 154 of file algorithm.h.

6.689.3.3 static RCPP_CONSTEXPR double Rcpp::algorithm::helpers::rtype_helper< double >::ZERO ( ) [inline], [static]

Definition at line 153 of file algorithm.h.

6.689.4 Member Data Documentation

6.689.4.1 RCPP_CONSTEXPR int Rcpp::algorithm::helpers::rtype_helper< double >::RTYPE = REALSXP [static]

Definition at line 151 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.690 Rcpp::algorithm::helpers::rtype_helper< int > Struct Template Reference

`#include <algorithm.h>`

Public Types

- typedef int type

Static Public Member Functions

- static int NA ()
- static RCPP_CONSTEXPR int ZERO ()
- static RCPP_CONSTEXPR int ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR int RTYPE = INTSXP

6.690.1 Detailed Description

template<>
struct Rcpp::algorithm::helpers::rtype_helper< int >

Definition at line 158 of file algorithm.h.

6.690.2 Member Typedef Documentation

6.690.2.1 typedef int Rcpp::algorithm::helpers::rtype_helper< int >::type

Definition at line 159 of file algorithm.h.

6.690.3 Member Function Documentation

6.690.3.1 static int Rcpp::algorithm::helpers::rtype_helper< int >::NA ( ) [inline], [static]

Definition at line 161 of file algorithm.h.

6.690.3.2 static RCPP_CONSTEXPR int Rcpp::algorithm::helpers::rtype_helper< int >::ONE ( ) [inline], [static]

Definition at line 163 of file algorithm.h.

6.690.3.3 static RCPP_CONSTEXPR int Rcpp::algorithm::helpers::rtype_helper< int >::ZERO ( ) [inline], [static]

Definition at line 162 of file algorithm.h.

6.690.4 Member Data Documentation

6.690.4.1 RCPP_CONSTEXPR int Rcpp::algorithm::helpers::rtype_helper< int >::RTYPE = INTSXP [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

6.691 Rcpp::rule Struct Reference

Public Attributes

- int r_type
- int r_day
- int r_week
- int r_mon
- long r_time

6.691.1 Detailed Description

Definition at line 349 of file Date.cpp.

6.691.2 Member Data Documentation

6.691.2.1 int Rcpp::rule::r_day

Definition at line 351 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.691.2.2 int Rcpp::rule::r_mon

Definition at line 353 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.691.2.3 long Rcpp::rule::r_time

Definition at line 354 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.691.2.4 int Rcpp::rule::r_type

Definition at line 350 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.691.2.5 int Rcpp::rule::r_week

Definition at line 352 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

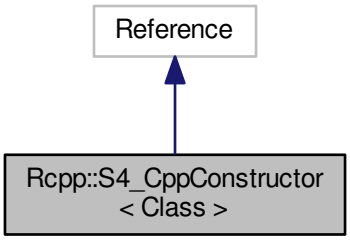
The documentation for this struct was generated from the following file:

- src/Date.cpp

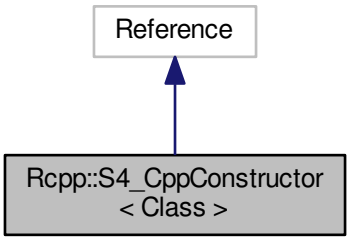
6.692 Rcpp::S4_CppConstructor< Class > Class Template Reference

`#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

6.692.1 Detailed Description

template<typename Class>
class Rcpp::S4_CppConstructor< Class >

Definition at line 203 of file Module.h.

6.692.2 Member Typedef Documentation

6.692.2.1 template<typename Class > typedef Reference Rcpp::S4_CppConstructor< Class >::Base [private]

Definition at line 204 of file Module.h.

6.692.2.2 template<typename Class > typedef Reference::Storage Rcpp::S4_CppConstructor< Class >::Storage

Definition at line 207 of file Module.h.

6.692.2.3 template<typename Class > typedef XPtr<class_Base> Rcpp::S4_CppConstructor< Class >::XP_Class

Definition at line 206 of file Module.h.

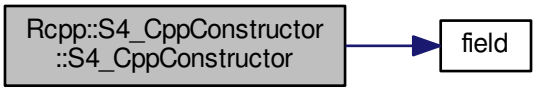
6.692.3 Constructor & Destructor Documentation

6.692.3.1 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 field(), RCPP_CTOR_ASSIGN_WITH_BASE, and RCPP_DEBUG.

Here is the call graph for this function:



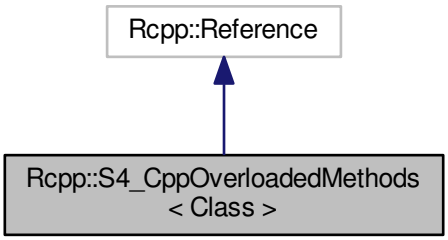
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

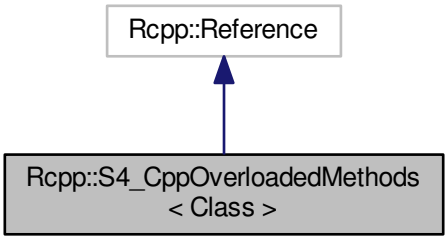
6.693 Rcpp::S4_CppOverloadedMethods< Class > Class Template Reference

`#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

6.693.1 Detailed Description

template<typename Class>
class Rcpp::S4_CppOverloadedMethods< Class >

Definition at line 224 of file Module.h.

6.693.2 Member Typedef Documentation

6.693.2.1 template<typename Class > typedef Rcpp::Reference Rcpp::S4_CppOverloadedMethods< Class >::Base [private]

Definition at line 225 of file Module.h.

6.693.2.2 template<typename Class > typedef SignedMethod<Class> Rcpp::S4_CppOverloadedMethods< Class >::signed_method_class

Definition at line 228 of file Module.h.

6.693.2.3 template<typename Class > typedef std::vector<signed_method_class*> Rcpp::S4_CppOverloaded↩ Methods< Class >::vec_signed_method

Definition at line 229 of file Module.h.

6.693.2.4 template<typename Class > typedef Rcpp::XPtr<class_Base> Rcpp::S4_CppOverloadedMethods< Class >::XP_Class

Definition at line 227 of file Module.h.

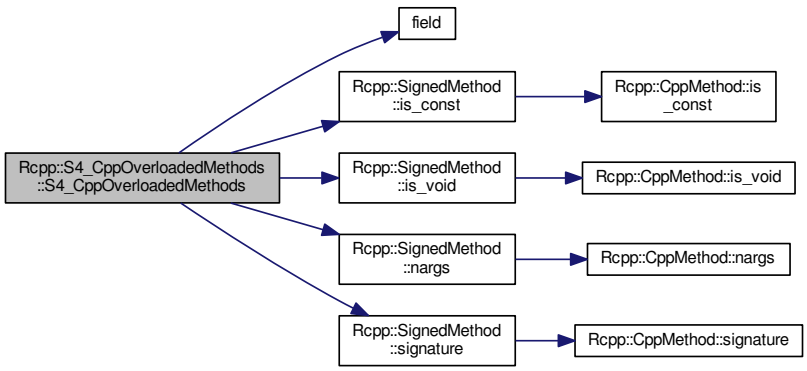
6.693.3 Constructor & Destructor Documentation

6.693.3.1 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(), RCPP_CTOR_ASSIGN_WITH_BASE, and Rcpp::SignedMethod< Class >::signature().

Here is the call graph for this function:



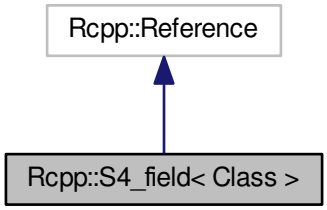
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

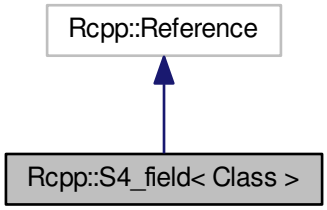
6.694 Rcpp::S4_field< Class > Class Template Reference

`#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 (CppProperty< Class > *p, const XP_Class &class_xp)

Private Types

- typedef Rcpp::Reference Base

6.694.1 Detailed Description

template<typename Class>
class Rcpp::S4_field< Class >

Definition at line 321 of file Module.h.

6.694.2 Member Typedef Documentation

6.694.2.1 template<typename Class > typedef Rcpp::Reference Rcpp::S4_field< Class >::Base [private]

Definition at line 322 of file Module.h.

6.694.2.2 template<typename Class > typedef XPtr<class_Base> Rcpp::S4_field< Class >::XP_Class

Definition at line 324 of file Module.h.

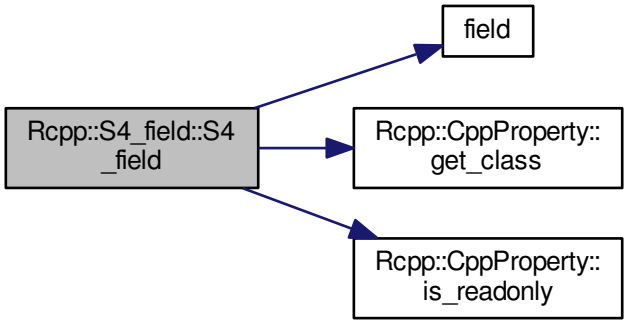
6.694.3 Constructor & Destructor Documentation

6.694.3.1 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 field(), Rcpp::CppProperty< Class >::get_class(), Rcpp::CppProperty< Class >::is_readonly(), RCPP_↩
CTOR_ASSIGN_WITH_BASE, and RCPP_DEBUG.

Here is the call graph for this function:



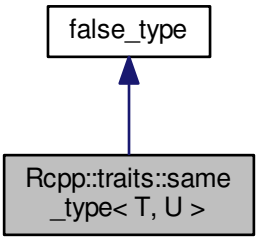
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

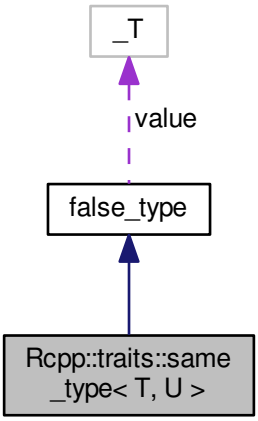
6.695 Rcpp::traits::same_type< T, U > Struct Template Reference

`#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

6.695.1 Detailed Description

template<typename T, typename U>
struct Rcpp::traits::same_type< T, U >

Definition at line 29 of file same_type.h.

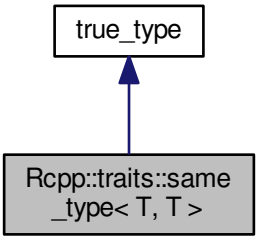
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/same_type.h

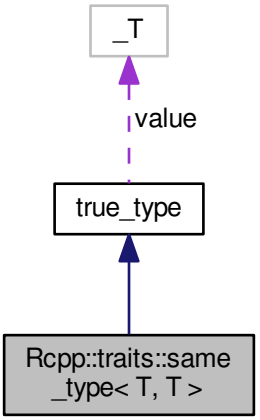
6.696 Rcpp::traits::same_type< T, T > Struct Template Reference

`#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

6.696.1 Detailed Description

template<typename T>
struct Rcpp::traits::same_type< T, T >

Definition at line 30 of file same_type.h.

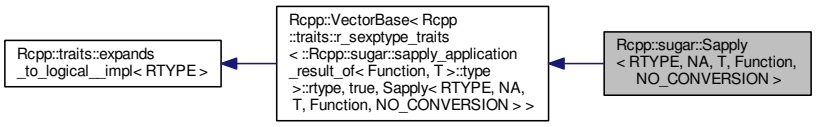
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/same_type.h

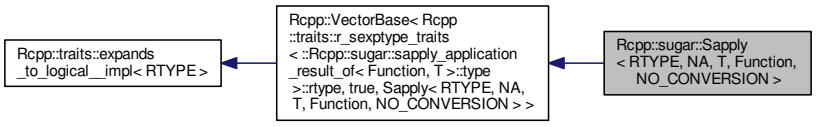
6.697 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION > Class Template Reference

`#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

6.697.1 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.

6.697.2 Member Typedef Documentation

6.697.2.1 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.

6.697.2.2 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.

6.697.2.3 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.

6.697.2.4 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.

6.697.2.5 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.

6.697.3 Constructor & Destructor Documentation

6.697.3.1 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.

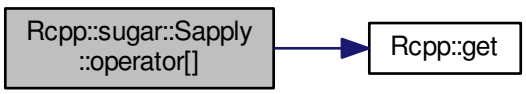
6.697.4 Member Function Documentation

6.697.4.1 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::get().

Here is the call graph for this function:



6.697.4.2 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.

6.697.5 Member Data Documentation

6.697.5.1 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.

6.697.5.2 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.

6.697.5.3 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.

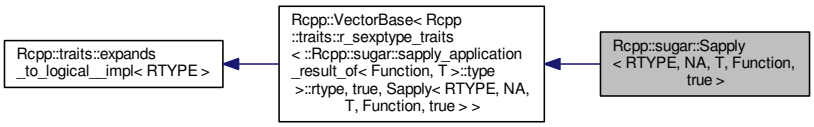
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sapply.h

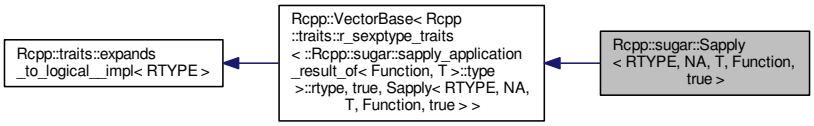
6.698 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true > Class Template Reference

`#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

6.698.1 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.

6.698.2 Member Typedef Documentation

6.698.2.1 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.

6.698.2.2 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.

6.698.2.3 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.

6.698.2.4 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.

6.698.3 Constructor & Destructor Documentation

6.698.3.1 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.

6.698.4 Member Function Documentation

6.698.4.1 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.

6.698.4.2 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.

6.698.5 Member Data Documentation

6.698.5.1 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.

6.698.5.2 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.

6.698.5.3 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

6.699 Rcpp::sugar::sapply_application_result_of< Function, SugarExpression > Struct Template Reference

```
#include <sapply.h>
```

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type type

6.699.1 Detailed Description

template<typename Function, typename SugarExpression>
struct Rcpp::sugar::sapply_application_result_of< Function, SugarExpression >

Definition at line 33 of file sapply.h.

6.699.2 Member Typedef Documentation

6.699.2.1 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

6.700 Rcpp::sugar::cbind_impl::scalar< RTYPE > Struct Template Reference

`#include <cbind.h>`

Public Types

- typedef cbind_storage_type< RTYPE >::type type

6.700.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::cbind_impl::scalar< RTYPE >

Definition at line 129 of file cbind.h.

6.700.2 Member Typedef Documentation

6.700.2.1 template<int RTYPE> typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::scalar< RTYPE >::type

Definition at line 130 of file cbind.h.

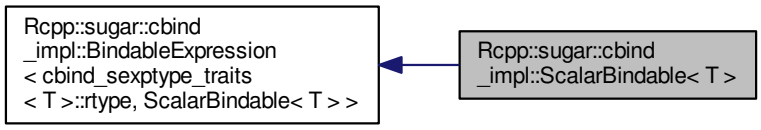
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

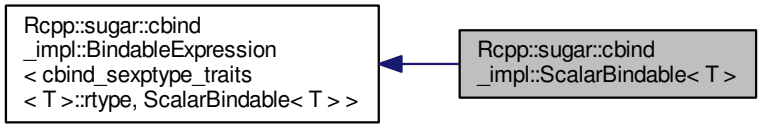
6.701 Rcpp::sugar::cbind_impl::ScalarBindable< T > Class Template Reference

`#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

6.701.1 Detailed Description

template<typename T>
class Rcpp::sugar::cbind_impl::ScalarBindable< T >

Definition at line 135 of file cbind.h.

6.701.2 Member Typedef Documentation

6.701.2.1 template<typename T > typedef T Rcpp::sugar::cbind_impl::ScalarBindable< T >::stored_type

Definition at line 139 of file cbind.h.

6.701.3 Member Enumeration Documentation

6.701.3.1 template<typename T > anonymous enum

Enumerator

RTYPE

Definition at line 140 of file cbind.h.

6.701.4 Constructor & Destructor Documentation

6.701.4.1 template<typename T > Rcpp::sugar::cbind_impl::ScalarBindable< T >::ScalarBindable (const T & *t_*) [inline]

Definition at line 146 of file cbind.h.

6.701.5 Member Function Documentation

6.701.5.1 template<typename T > R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::ncol ( ) const [inline]

Definition at line 152 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::ncol().

6.701.5.2 template<typename T > R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::nrow ( ) const [inline]

Definition at line 150 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::nrow(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, Scalar↩ Bindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >↩ ::operator[]() , and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::size().

6.701.5.3 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.

6.701.5.4 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.

6.701.5.5 template<typename T > R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::size ( ) const [inline]

Definition at line 148 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::size().

6.701.6 Member Data Documentation

6.701.6.1 template<typename T > T Rcpp::sugar::cbind_impl::ScalarBindable< T >::t [private]

Definition at line 143 of file cbind.h.

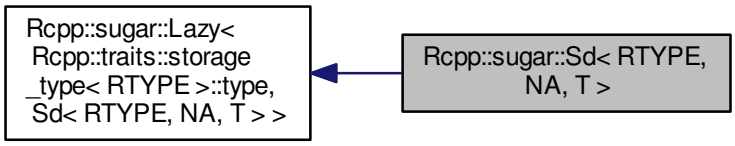
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

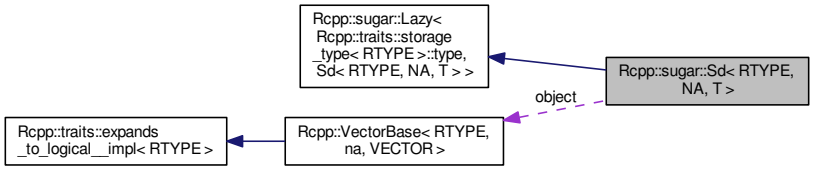
6.702 Rcpp::sugar::Sd< RTYPE, NA, T > Class Template Reference

`#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

6.702.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sd< RTYPE, NA, T >

Definition at line 29 of file sd.h.

6.702.2 Member Typedef Documentation

6.702.2.1 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.

6.702.2.2 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.

6.702.3 Constructor & Destructor Documentation

6.702.3.1 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.

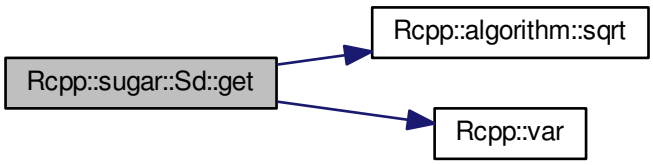
6.702.4 Member Function Documentation

6.702.4.1 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:



6.702.5 Member Data Documentation

6.702.5.1 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

6.703 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)
- int get_index (STORAGE value) const
- int get_addr (STORAGE value) const
- template<>
  int get_addr (int value) const
- template<>
  int get_addr (double val) const
- template<>
  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_

6.703.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::SelfHash< RTYPE >

Definition at line 32 of file SelfHash.h.

6.703.2 Member Typedef Documentation

6.703.2.1 template<int RTYPE> typedef traits::storage_type<RTYPE>::type Rcpp::sugar::SelfHash< RTYPE >::STORAGE

Definition at line 34 of file SelfHash.h.

6.703.2.2 template<int RTYPE> typedef Vector<RTYPE> Rcpp::sugar::SelfHash< RTYPE >::VECTOR

Definition at line 35 of file SelfHash.h.

6.703.3 Constructor & Destructor Documentation

6.703.3.1 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.

6.703.4 Member Function Documentation

6.703.4.1 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 >::get_addr(), and Rcpp::sugar::SelfHash< RTYPE >::size_.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match().

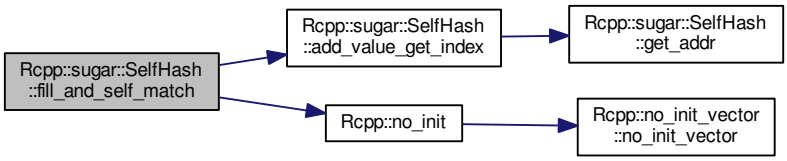
Here is the call graph for this function:

6.703.4.2 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().

Here is the call graph for this function:



6.703.4.3 template<int RTYPE> 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().

6.703.4.4 template<> int Rcpp::sugar::SelfHash< INTSXP >::get_addr (int *value*) const [inline]

Definition at line 94 of file SelfHash.h.

References RCPP_HASH.

6.703.4.5 template<> int Rcpp::sugar::SelfHash< REALSXP >::get_addr (double *val*) const [inline]

Definition at line 98 of file SelfHash.h.

References RCPP_HASH.

6.703.4.6 template<> int Rcpp::sugar::SelfHash< STRSXP >::get_addr (SEXP *value*) const [inline]

Definition at line 115 of file SelfHash.h.

References RCPP_HASH.

6.703.4.7 template<int RTYPE> int Rcpp::sugar::SelfHash< RTYPE >::get_index (STORAGE *value*) const [inline]

Definition at line 78 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::get_addr().

Here is the call graph for this function:

6.703.4.8 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_.

6.703.5 Member Data Documentation

6.703.5.1 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 >::SelfHash().

6.703.5.2 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 >::SelfHash().

6.703.5.3 template<int RTYPE> int Rcpp::sugar::SelfHash< RTYPE >::k

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.703.5.4 template<int RTYPE> int Rcpp::sugar::SelfHash< RTYPE >::m

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.703.5.5 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().

6.703.5.6 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().

6.703.5.7 template<int RTYPE> STORAGE* Rcpp::sugar::SelfHash< RTYPE >::src

Definition at line 58 of file SelfHash.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/hash/SelfHash.h

6.704 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

6.704.1 Detailed Description

template<typename HASH, typename STORAGE>
class Rcpp::sugar::SelfInserter< HASH, STORAGE >

Definition at line 29 of file self_match.h.

6.704.2 Constructor & Destructor Documentation

6.704.2.1 template<typename HASH , typename STORAGE > Rcpp::sugar::SelfInserter< HASH, STORAGE >::SelfInserter (HASH & *hash_*) [inline]

Definition at line 31 of file self_match.h.

6.704.3 Member Function Documentation

6.704.3.1 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.

6.704.4 Member Data Documentation

6.704.4.1 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()(), and Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch().

6.704.4.2 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()().

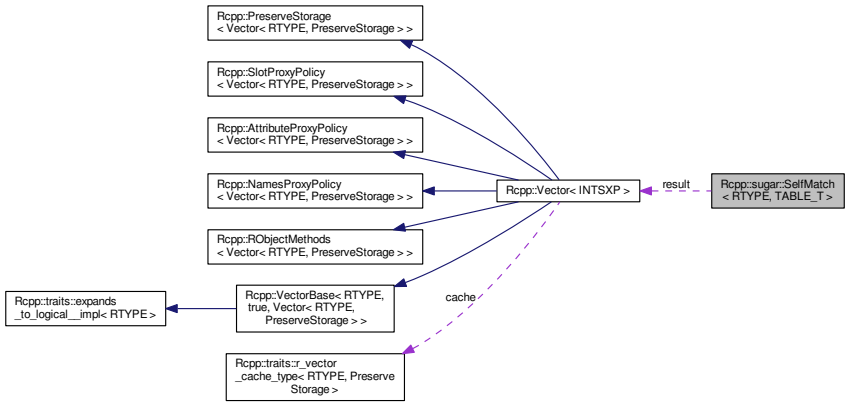
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self_match.h

6.705 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

6.705.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >

Definition at line 49 of file self_match.h.

6.705.2 Member Typedef Documentation

6.705.2.1 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.

6.705.2.2 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.

6.705.2.3 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.

6.705.3 Constructor & Destructor Documentation

6.705.3.1 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::sugar::SelfInserter< HASH, STORAGE >::hash.

6.705.4 Member Function Documentation

6.705.4.1 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.

6.705.5 Member Data Documentation

6.705.5.1 template<int RTYPE, typename TABLE_T > HASH Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::hash [private]

Definition at line 62 of file self_match.h.

6.705.5.2 template<int RTYPE, typename TABLE_T > IntegerVector Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::result [private]

Definition at line 63 of file self_match.h.

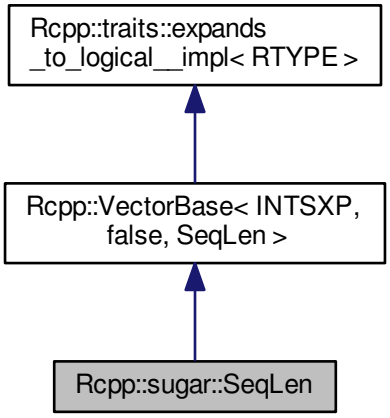
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self_match.h

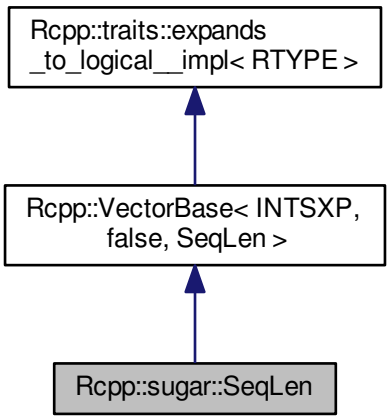
6.706 Rcpp::sugar::SeqLen Class Reference

```
#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

6.706.1 Detailed Description

Definition at line 28 of file seq_along.h.

6.706.2 Constructor & Destructor Documentation

6.706.2.1 Rcpp::sugar::SeqLen::SeqLen (R_xlen_t *len_*) [inline]

Definition at line 30 of file seq_along.h.

6.706.3 Member Function Documentation

6.706.3.1 R_xlen_t Rcpp::sugar::SeqLen::operator[ ] (R_xlen_t *i*) const [inline]

Definition at line 32 of file seq_along.h.

6.706.3.2 R_xlen_t Rcpp::sugar::SeqLen::size ( ) const [inline]

Definition at line 35 of file seq_along.h.

References len.

6.706.4 Member Data Documentation

6.706.4.1 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

6.707 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

6.707.1 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.

6.707.2 Member Typedef Documentation

6.707.2.1 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.

6.707.2.2 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.

6.707.2.3 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.

6.707.3 Constructor & Destructor Documentation

6.707.3.1 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.

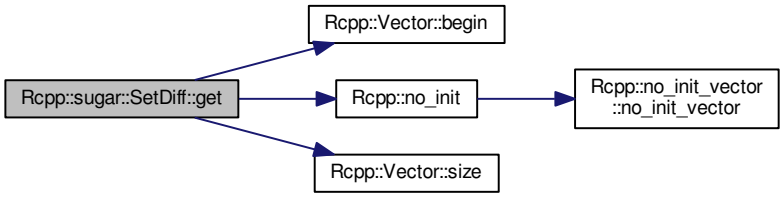
6.707.4 Member Function Documentation

6.707.4.1 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::no_init(), and Rcpp::Vector< RTYPE, Storage↩ Policy >::size().

Here is the call graph for this function:



6.707.5 Member Data Documentation

6.707.5.1 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.

6.707.5.2 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

6.708 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

6.708.1 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.

6.708.2 Member Typedef Documentation

6.708.2.1 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.

6.708.2.2 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.

6.708.2.3 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.

6.708.3 Constructor & Destructor Documentation

6.708.3.1 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.

6.708.4 Member Function Documentation

6.708.4.1 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.

6.708.5 Member Data Documentation

6.708.5.1 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.

6.708.5.2 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

6.709 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

6.709.1 Detailed Description

template<class T>
class Rcpp::Shelter< T >

Definition at line 24 of file Shelter.h.

6.709.2 Constructor & Destructor Documentation

6.709.2.1 template<class T > Rcpp::Shelter< T >::Shelter ( ) [inline]

Definition at line 26 of file Shelter.h.

6.709.2.2 template<class T > Rcpp::Shelter< T >::~Shelter ( ) [inline]

Definition at line 33 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected.

6.709.2.3 template<class T > Rcpp::Shelter< T >::Shelter (const Shelter< T > &) [private]

6.709.3 Member Function Documentation

6.709.3.1 template<class T > SEXP Rcpp::Shelter< T >::operator() (SEXP *x*) [inline]

Definition at line 28 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected.

6.709.3.2 template<class T > Shelter& Rcpp::Shelter< T >::operator= (const Shelter< T > &) [private]

6.709.4 Member Data Documentation

6.709.4.1 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

6.710 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 &)

6.710.1 Detailed Description

template<typename T>
class Rcpp::Shield< T >

Definition at line 29 of file Shield.h.

6.710.2 Constructor & Destructor Documentation

6.710.2.1 template<typename T> Rcpp::Shield< T >::Shield (SEXP t_) [inline]

Definition at line 31 of file Shield.h.

6.710.2.2 template<typename T> Rcpp::Shield< T >::~Shield ( ) [inline]

Definition at line 32 of file Shield.h.

References Rcpp::Shield< T >::t.

6.710.2.3 template<typename T> Rcpp::Shield< T >::Shield (const Shield< T > &) [private]

6.710.3 Member Function Documentation

6.710.3.1 template<typename T> Rcpp::Shield< T >::operator SEXP ( ) const [inline]

Definition at line 36 of file Shield.h.

References Rcpp::Shield< T >::t.

6.710.3.2 template<typename T> Shield& Rcpp::Shield< T >::operator= (const Shield< T > &) [private]

6.710.4 Member Data Documentation

6.710.4.1 template<typename T> SEXP Rcpp::Shield< T >::t

Definition at line 37 of file Shield.h.

Referenced by Rcpp::Shield< T >::operator SEXP(), and Rcpp::Shield< T >::~Shield().

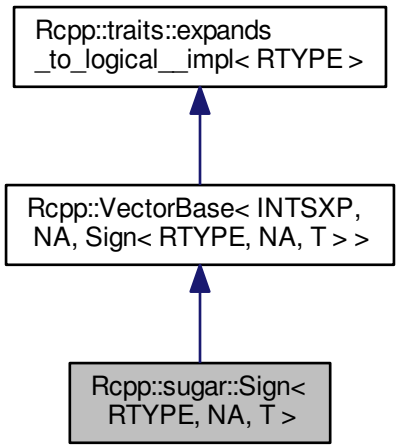
The documentation for this class was generated from the following file:

- inst/include/Rcpp/protection/Shield.h

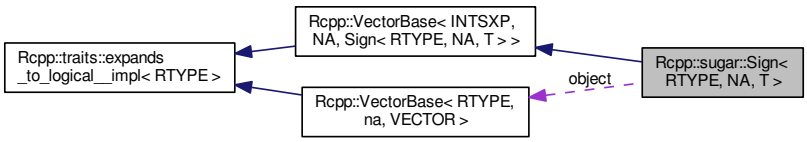
6.711 Rcpp::sugar::Sign< RTYPE, NA, T > Class Template Reference

`#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

6.711.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sign< RTYPE, NA, T >

Definition at line 48 of file sign.h.

6.711.2 Member Typedef Documentation

6.711.2.1 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.

6.711.2.2 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.

6.711.2.3 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.

6.711.3 Constructor & Destructor Documentation

6.711.3.1 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.

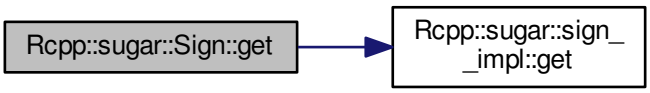
6.711.4 Member Function Documentation

6.711.4.1 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().

Here is the call graph for this function:

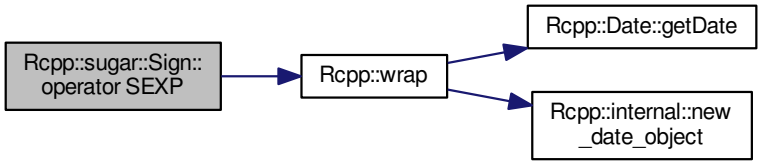


6.711.4.2 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP ( ) const [inline]

Definition at line 61 of file sign.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.711.4.3 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.

6.711.4.4 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.

6.711.5 Member Data Documentation

6.711.5.1 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

6.712 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)

6.712.1 Detailed Description

template<bool NA, int RTYPE>
class Rcpp::sugar::sign__impl< NA, RTYPE >

Definition at line 29 of file sign.h.

6.712.2 Member Typedef Documentation

6.712.2.1 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.

6.712.3 Member Function Documentation

6.712.3.1 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

6.713 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)

6.713.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::sign__impl< false, RTYPE >

Definition at line 38 of file sign.h.

6.713.2 Member Typedef Documentation

6.713.2.1 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.

6.713.3 Member Function Documentation

6.713.3.1 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

6.714 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

6.714.1 Detailed Description

template<typename Class>
class Rcpp::SignedConstructor< Class >

Definition at line 143 of file Module.h.

6.714.2 Constructor & Destructor Documentation

6.714.2.1 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.

6.714.3 Member Function Documentation

6.714.3.1 template<typename Class> int Rcpp::SignedConstructor< Class >::nargs ( ) [inline]

Definition at line 156 of file Module.h.

6.714.3.2 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.

6.714.4 Member Data Documentation

6.714.4.1 template<typename Class> Constructor_Base<Class>* Rcpp::SignedConstructor< Class >::ctor

Definition at line 152 of file Module.h.

6.714.4.2 template<typename Class> std::string Rcpp::SignedConstructor< Class >::docstring

Definition at line 154 of file Module.h.

6.714.4.3 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

6.715 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

6.715.1 Detailed Description

template<typename Class>
class Rcpp::SignedFactory< Class >

Definition at line 163 of file Module.h.

6.715.2 Constructor & Destructor Documentation

6.715.2.1 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.

6.715.3 Member Function Documentation

6.715.3.1 template<typename Class > int Rcpp::SignedFactory< Class >::nargs ( ) [inline]

Definition at line 176 of file Module.h.

6.715.3.2 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.

6.715.4 Member Data Documentation

6.715.4.1 template<typename Class > std::string Rcpp::SignedFactory< Class >::docstring

Definition at line 174 of file Module.h.

6.715.4.2 template<typename Class > Factory_Base<Class>* Rcpp::SignedFactory< Class >::fact

Definition at line 172 of file Module.h.

6.715.4.3 template<typename Class > ValidConstructor Rcpp::SignedFactory< Class >::valid

Definition at line 173 of file Module.h.

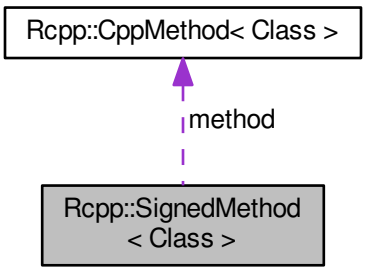
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.716 Rcpp::SignedMethod< Class > 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

6.716.1 Detailed Description

template<typename Class>
class Rcpp::SignedMethod< Class >

Definition at line 184 of file Module.h.

6.716.2 Member Typedef Documentation

6.716.2.1 template<typename Class > typedef CppMethod<Class> Rcpp::SignedMethod< Class >::METHOD

Definition at line 186 of file Module.h.

6.716.3 Constructor & Destructor Documentation

6.716.3.1 template<typename Class > Rcpp::SignedMethod< Class >::SignedMethod (METHOD * *m*, ValidMethod *valid_*, const char * *doc*) [inline]

Definition at line 187 of file Module.h.

6.716.4 Member Function Documentation

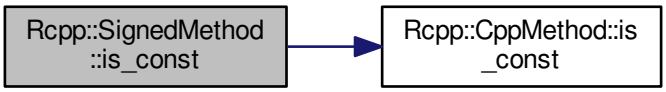
6.716.4.1 template<typename Class > bool Rcpp::SignedMethod< Class >::is_const ( ) [inline]

Definition at line 195 of file Module.h.

References Rcpp::CppMethod< Class >::is_const().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



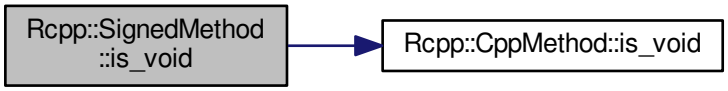
6.716.4.2 template<typename Class > bool Rcpp::SignedMethod< Class >::is_void ( ) [inline]

Definition at line 194 of file Module.h.

References Rcpp::CppMethod< Class >::is_void().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



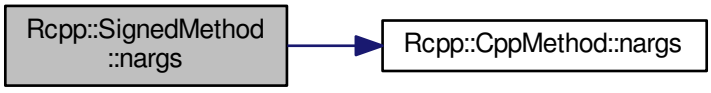
6.716.4.3 template<typename Class > int Rcpp::SignedMethod< Class >::nargs ( ) [inline]

Definition at line 193 of file Module.h.

References Rcpp::CppMethod< Class >::nargs().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



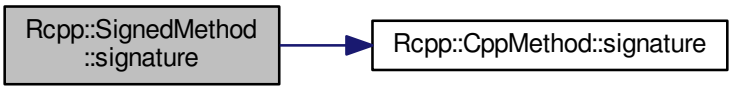
6.716.4.4 template<typename Class > void Rcpp::SignedMethod< Class >::signature (std::string & *s*, const char * *name*) [inline]

Definition at line 196 of file Module.h.

References Rcpp::CppMethod< Class >::signature().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



6.716.5 Member Data Documentation

6.716.5.1 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().

6.716.5.2 template<typename Class > METHOD* Rcpp::SignedMethod< Class >::method

Definition at line 189 of file Module.h.

6.716.5.3 template<typename Class > ValidMethod Rcpp::SignedMethod< Class >::valid

Definition at line 190 of file Module.h.

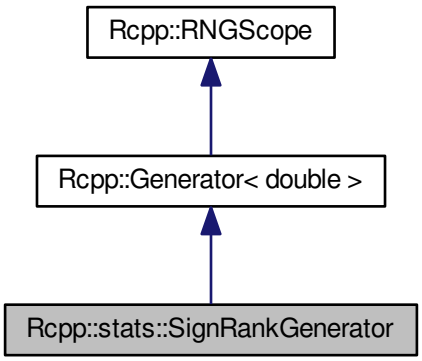
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

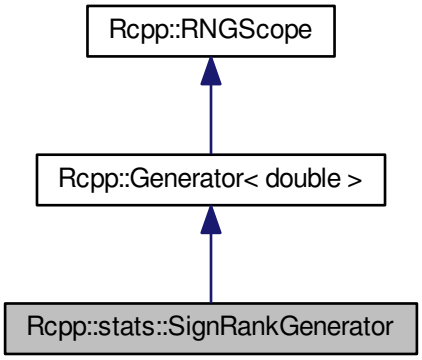
6.717 Rcpp::stats::SignRankGenerator Class Reference

`#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

6.717.1 Detailed Description

Definition at line 29 of file rsignrank.h.

6.717.2 Constructor & Destructor Documentation

6.717.2.1 Rcpp::stats::SignRankGenerator::SignRankGenerator (double *nn_*) [inline]

Definition at line 31 of file rsignrank.h.

6.717.3 Member Function Documentation

6.717.3.1 double Rcpp::stats::SignRankGenerator::operator() ( ) const [inline]

Definition at line 32 of file rsignrank.h.

References nn.

6.717.4 Member Data Documentation

6.717.4.1 double Rcpp::stats::SignRankGenerator::nn [private]

Definition at line 34 of file rsignrank.h.

Referenced by operator()().

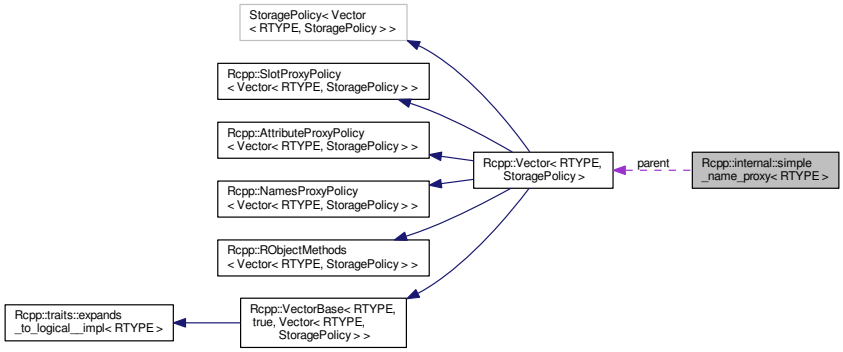
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rsignrank.h

6.718 Rcpp::internal::simple_name_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Collaboration diagram for Rcpp::internal::simple_name_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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

6.718.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::simple_name_proxy< RTYPE >

Definition at line 32 of file 00_forward_proxy.h.

6.718.2 Member Typedef Documentation

6.718.2.1 template<int RTYPE> typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::simple_name_proxy< RTYPE >::CTYPE

Definition at line 31 of file proxy.h.

6.718.2.2 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::VECTOR

Definition at line 30 of file proxy.h.

6.718.3 Constructor & Destructor Documentation

6.718.3.1 template<int RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::simple_name_proxy (VECTOR & *v*, const std::string & *name_*) [inline]

Definition at line 32 of file proxy.h.

6.718.3.2 template<int RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::simple_name_proxy (const simple_name_proxy< RTYPE > & *other*) [inline]

Definition at line 34 of file proxy.h.

6.718.3.3 template<int RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::~simple_name_proxy ( ) [inline]

Definition at line 36 of file proxy.h.

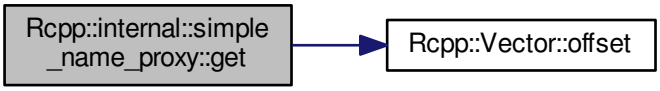
6.718.4 Member Function Documentation

6.718.4.1 template<int RTYPE> CTYPE Rcpp::internal::simple_name_proxy< RTYPE >::get ( ) const [inline], [private]

Definition at line 76 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



6.718.4.2 template<int RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::operator CTYPE ( ) const [inline]

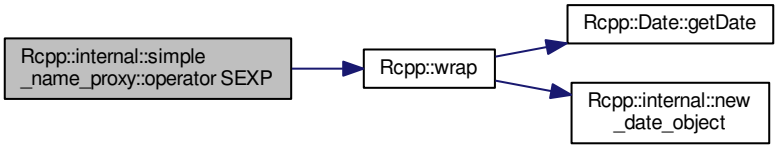
Definition at line 55 of file proxy.h.

6.718.4.3 template<int RTYPE> Rcpp::internal::simple_name_proxy< RTYPE >::operator SEXP ( ) const [inline]

Definition at line 60 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.718.4.4 template<int RTYPE> simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE >::operator= (CTYPE *rhs*) [inline]

Definition at line 38 of file proxy.h.

6.718.4.5 template<int RTYPE> simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE >::operator= (const simple_name_proxy< RTYPE > & *other*) [inline]

Definition at line 42 of file proxy.h.

6.718.4.6 template<int RTYPE> template<typename T > simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE >::operator= (const T & *rhs*) [inline]

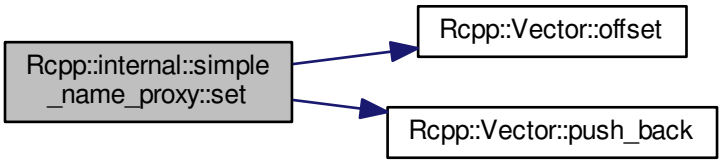
Definition at line 48 of file proxy.h.

6.718.4.7 template<int RTYPE> void Rcpp::internal::simple_name_proxy< RTYPE >::set (CTYPE *rhs*) [inline], [private]

Definition at line 67 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Here is the call graph for this function:



6.718.5 Member Data Documentation

6.718.5.1 template<int RTYPE> std::string Rcpp::internal::simple_name_proxy< RTYPE >::name [private]

Definition at line 66 of file proxy.h.

6.718.5.2 template<int RTYPE> VECTOR& Rcpp::internal::simple_name_proxy< RTYPE >::parent [private]

Definition at line 65 of file proxy.h.

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

6.719 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

6.719.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::SingleLogicalResult< NA, T >

Definition at line 42 of file SingleLogicalResult.h.

6.719.2 Constructor & Destructor Documentation

6.719.2.1 template<bool NA, typename T> Rcpp::sugar::SingleLogicalResult< NA, T >::SingleLogicalResult ( ) [inline]

Definition at line 46 of file SingleLogicalResult.h.

6.719.3 Member Function Documentation

6.719.3.1 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::apply ( ) [inline]

Definition at line 48 of file SingleLogicalResult.h.

6.719.3.2 template<bool NA, typename T> int Rcpp::sugar::SingleLogicalResult< NA, T >::get ( ) [inline]

Definition at line 81 of file SingleLogicalResult.h.

Referenced by 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::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, 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::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), and Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

6.719.3.3 template<bool NA, typename T> SEXP Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp ( ) [inline]

Definition at line 86 of file SingleLogicalResult.h.

6.719.3.4 template<bool NA, typename T> bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_false ( ) [inline]

Definition at line 59 of file SingleLogicalResult.h.

6.719.3.5 template<bool NA, typename T> bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_na ( ) [inline]

Definition at line 64 of file SingleLogicalResult.h.

6.719.3.6 template<bool NA, typename T> bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_true ( ) [inline]

Definition at line 54 of file SingleLogicalResult.h.

6.719.3.7 template<bool NA, typename T> bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_unresolved ( ) [inline], [protected]

Definition at line 98 of file SingleLogicalResult.h.

6.719.3.8 template<bool NA, typename T> Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool ( ) [inline]

Definition at line 73 of file SingleLogicalResult.h.

6.719.3.9 template<bool NA, typename T> Rcpp::sugar::SingleLogicalResult< NA, T >::operator SEXP ( ) [inline]

Definition at line 69 of file SingleLogicalResult.h.

6.719.3.10 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::reset ( ) [inline], [protected]

Definition at line 94 of file SingleLogicalResult.h.

6.719.3.11 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::set (int *x*) [inline], [protected]

Definition at line 93 of file SingleLogicalResult.h.

6.719.3.12 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::set_false ( ) [inline], [protected]

Definition at line 96 of file SingleLogicalResult.h.

6.719.3.13 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::set_na ( ) [inline], [protected]

Definition at line 97 of file SingleLogicalResult.h.

6.719.3.14 template<bool NA, typename T> void Rcpp::sugar::SingleLogicalResult< NA, T >::set_true ( ) [inline], [protected]

Definition at line 95 of file SingleLogicalResult.h.

6.719.3.15 template<bool NA, typename T> int Rcpp::sugar::SingleLogicalResult< NA, T >::size ( ) [inline]

Definition at line 79 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::And_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::And_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::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size().

6.719.4 Member Data Documentation

6.719.4.1 template<bool NA, typename T> int Rcpp::sugar::SingleLogicalResult< NA, T >::result [protected]

Definition at line 92 of file SingleLogicalResult.h.

6.719.4.2 template<bool NA, typename T> const int Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED = -5 [static]

Definition at line 44 of file SingleLogicalResult.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

6.720 Rcpp::SingleLogicalResult< NA, T > Class Template Reference

`#include <Vector.h>`

6.720.1 Detailed Description

template<bool NA, typename T>
class Rcpp::SingleLogicalResult< NA, T >

Definition at line 40 of file Vector.h.

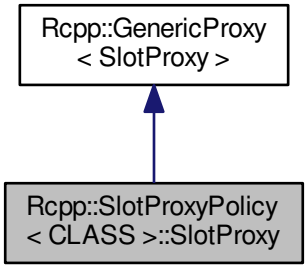
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Vector.h

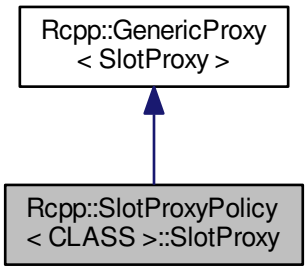
6.721 Rcpp::SlotProxyPolicy< CLASS >::SlotProxy Class Reference

`#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

6.721.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >::SlotProxy

Definition at line 27 of file SlotProxy.h.

6.721.2 Constructor & Destructor Documentation

6.721.2.1 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.

Referenced by Rcpp::SlotProxyPolicy< Vector< RTYPE, StoragePolicy > >::slot().

6.721.3 Member Function Documentation

6.721.3.1 template<typename CLASS> SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get ( ) const [inline], [private]

Definition at line 51 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=().

6.721.3.2 template<typename CLASS> Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator SEXP ( ) const [inline]

Definition at line 43 of file SlotProxy.h.

6.721.3.3 template<typename CLASS > template<typename T > Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator T ( ) const

Definition at line 91 of file proxy.h.

6.721.3.4 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().

Here is the call graph for this function:

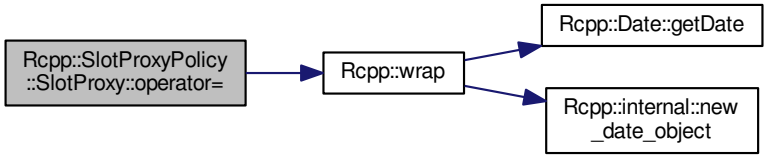
6.721.3.5 template<typename CLASS> template<typename T > SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (const T & *rhs*)

6.721.3.6 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:



6.721.3.7 template<typename CLASS> void Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::set (SEXP *x*) [inline], [private]

Definition at line 54 of file SlotProxy.h.

6.721.4 Member Data Documentation

6.721.4.1 template<typename CLASS> CLASS& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent [private]

Definition at line 48 of file SlotProxy.h.

6.721.4.2 template<typename CLASS> SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot_name [private]

Definition at line 49 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::const_SlotProxy(), 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

6.722 Rcpp::SlotProxyPolicy< CLASS > Class Template Reference

`#include <SlotProxy.h>`

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

6.722.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >

Definition at line 24 of file SlotProxy.h.

6.722.2 Member Function Documentation

6.722.2.1 template<typename CLASS> bool Rcpp::SlotProxyPolicy< CLASS >::hasSlot (const std::string & *name*) const [inline]

Definition at line 94 of file SlotProxy.h.

6.722.2.2 template<typename CLASS> SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (const std::string & *name*) [inline]

Definition at line 83 of file SlotProxy.h.

6.722.2.3 template<typename CLASS> const_SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (const std::string & *name*) const [inline]

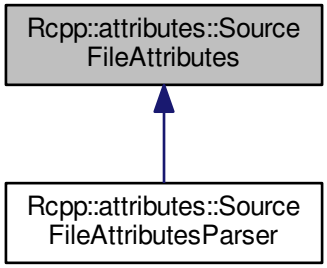
Definition at line 88 of file SlotProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/SlotProxy.h

6.723 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

6.723.1 Detailed Description

Definition at line 397 of file attributes.cpp.

6.723.2 Member Typedef Documentation

6.723.2.1 typedef std::vector<Attribute>::const_iterator Rcpp::attributes::SourceFileAttributes::const_iterator

Definition at line 404 of file attributes.cpp.

6.723.3 Constructor & Destructor Documentation

6.723.3.1 virtual Rcpp::attributes::SourceFileAttributes::~SourceFileAttributes ( ) [inline], [virtual]

Definition at line 400 of file attributes.cpp.

6.723.4 Member Function Documentation

6.723.4.1 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::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::generateCpp(), and Rcpp::attributes::isRoxygenCpp().

6.723.4.2 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::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::generateCpp(), and Rcpp::attributes::isRoxygenCpp().

6.723.4.3 virtual bool Rcpp::attributes::SourceFileAttributes::hasGeneratorOutput ( ) const [pure virtual]

Implemented in Rcpp::attributes::SourceFileAttributesParser.

6.723.4.4 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::RExportsGenerator::doWriteFunctions(), and Rcpp::attributes::ExportsGenerator::writeFunctions().

6.723.4.5 virtual const std::vector<std::string>& Rcpp::attributes::SourceFileAttributes::modules ( ) const [pure virtual]

Implemented in Rcpp::attributes::SourceFileAttributesParser.

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

6.723.4.6 virtual const std::vector<std::vector<std::string> >& Rcpp::attributes::SourceFileAttributes::roxygenChunks ( ) const [pure virtual]

Implemented in Rcpp::attributes::SourceFileAttributesParser.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

6.723.4.7 virtual const std::string& Rcpp::attributes::SourceFileAttributes::sourceFile ( ) const [pure virtual]

Implemented in Rcpp::attributes::SourceFileAttributesParser.

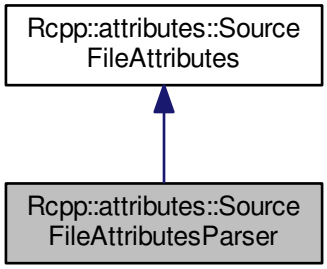
Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

The documentation for this class was generated from the following file:

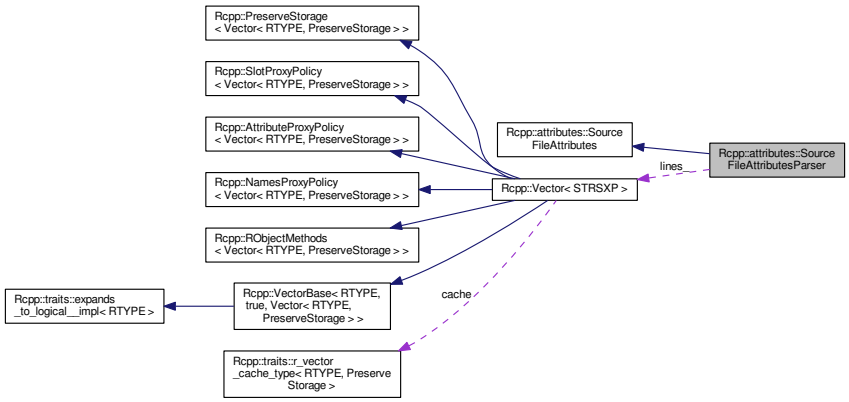
- src/attributes.cpp

6.724 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, bool parseDependencies)
- 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
- 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_
- std::vector< std::string > embeddedR_
- std::vector< FileInfo > sourceDependencies_
- std::vector< std::vector< std::string > > roxygenChunks_
- std::vector< std::string > roxygenBuffer_

Additional Inherited Members

6.724.1 Detailed Description

Definition at line 445 of file attributes.cpp.

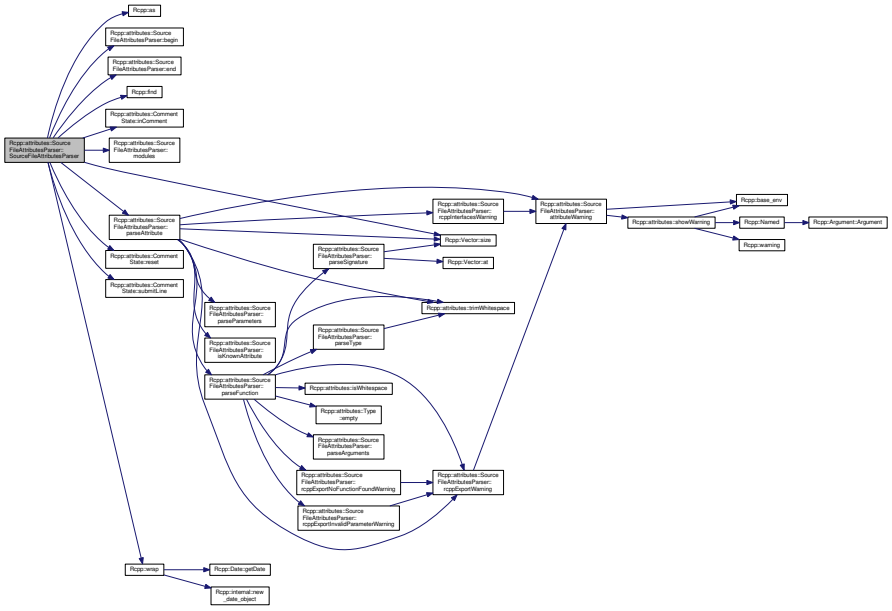
6.724.2 Constructor & Destructor Documentation

6.724.2.1 Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (const std::string & *sourceFile*, bool *parseDependencies*) [explicit]

Definition at line 1117 of file attributes.cpp.

References Rcpp::as(), attributes_, begin(), embeddedR_, end(), Rcpp::find(), Rcpp::attributes::CommentState::inComment(), lines_, modules(), modules_, parseAttribute(), Rcpp::attributes::CommentState::reset(), roxygenBuffer_, roxygenChunks_, Rcpp::Vector< RTYPE, StoragePolicy >::size(), sourceDependencies_, sourceFile_, Rcpp::attributes::CommentState::submitLine(), and Rcpp::wrap().

Here is the call graph for this function:



6.724.2.2 Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (const SourceFileAttributesParser &) [private]

6.724.3 Member Function Documentation

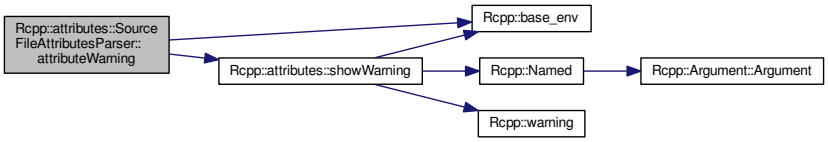
6.724.3.1 void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (const std::string & *message,* const std::string & *attribute,* size_t *lineNumber*) [private]

Definition at line 1632 of file attributes.cpp.

References Rcpp::base_env(), Rcpp::attributes::showWarning(), and sourceFile_.

Referenced by attributeWarning(), parseAttribute(), rcppExportWarning(), and rcppInterfacesWarning().

Here is the call graph for this function:

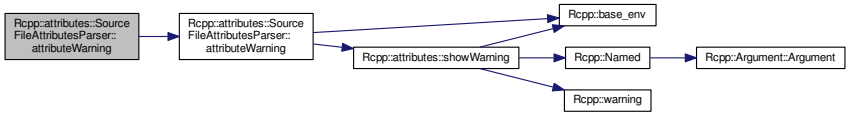


6.724.3.2 void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (const std::string & *message*, size_t *lineNumber*) [private]

Definition at line 1650 of file attributes.cpp.

References attributeWarning().

Here is the call graph for this function:



6.724.3.3 virtual const_iterator Rcpp::attributes::SourceFileAttributesParser::begin ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 460 of file attributes.cpp.

Referenced by compileAttributes(), Rcpp::attributes::isRoxygenCpp(), and SourceFileAttributesParser().

6.724.3.4 const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::embeddedR ( ) const [inline]

Definition at line 495 of file attributes.cpp.

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

6.724.3.5 virtual const_iterator Rcpp::attributes::SourceFileAttributesParser::end ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 461 of file attributes.cpp.

Referenced by compileAttributes(), Rcpp::attributes::isRoxygenCpp(), and SourceFileAttributesParser().

6.724.3.6 virtual bool Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 472 of file attributes.cpp.

Referenced by compileAttributes().

6.724.3.7 virtual bool Rcpp::attributes::SourceFileAttributesParser::hasInterface (const std::string & *name*) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 479 of file attributes.cpp.

References Rcpp::attributes::kInterfaceR, and Rcpp::attributes::kInterfacesAttribute.

6.724.3.8 bool Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute (const std::string & *name*) const [private]

Definition at line 1623 of file attributes.cpp.

References Rcpp::attributes::kDependsAttribute, Rcpp::attributes::kExportAttribute, Rcpp::attributes::kInterfaces↩Attribute, and Rcpp::attributes::kPluginsAttribute.

Referenced by parseAttribute().

6.724.3.9 virtual const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::modules ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 463 of file attributes.cpp.

Referenced by Rcpp::attributes::isRoxygenCpp(), and SourceFileAttributesParser().

6.724.3.10 SourceFileAttributesParser& Rcpp::attributes::SourceFileAttributesParser::operator= (const SourceFileAttributesParser &) [private]

6.724.3.11 std::vector< std::string > Rcpp::attributes::SourceFileAttributesParser::parseArguments (const std::string & *argText*) [private]

Definition at line 1534 of file attributes.cpp.

Referenced by parseFunction().

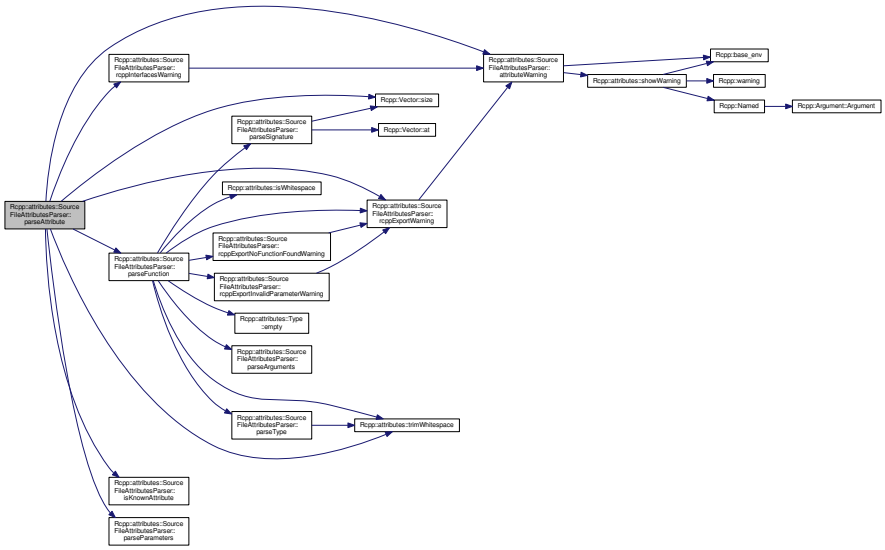
6.724.3.12 Attribute Rcpp::attributes::SourceFileAttributesParser::parseAttribute (const std::vector< std::string > & *match,* int *lineNumber*) [private]

Definition at line 1243 of file attributes.cpp.

References attributeWarning(), isKnownAttribute(), Rcpp::attributes::kExportAttribute, Rcpp::attributes::kExportName, Rcpp::attributes::kExportRng, Rcpp::attributes::kInterfaceCpp, Rcpp::attributes::kInterfaceR, Rcpp::attributes::kInterfacesAttribute, Rcpp::attributes::kParamValueFalse, Rcpp::attributes::kParamValueFALSE, Rcpp::attributes::kParamValueTrue, Rcpp::attributes::kParamValueTRUE, lines_, parseFunction(), parseParameters(), rcppExportWarning(), rcppInterfacesWarning(), roxygenBuffer_, Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::attributes::trimWhitespace().

Referenced by SourceFileAttributesParser().

Here is the call graph for this function:



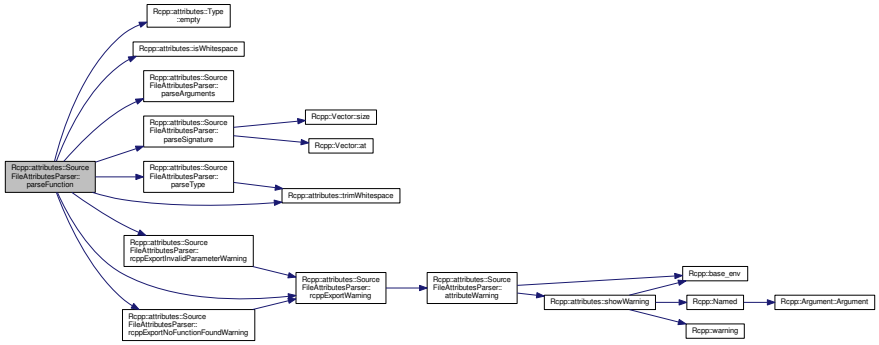
6.724.3.13 Function Rcpp::attributes::SourceFileAttributesParser::parseFunction (size_t *lineNumber*) [private]

Definition at line 1365 of file attributes.cpp.

References Rcpp::attributes::Type::empty(), Rcpp::attributes::isWhitespace(), Rcpp::attributes::kWhitespaceChars, parseArguments(), parseSignature(), parseType(), rcppExportInvalidParameterWarning(), rcppExportNoFunctionFoundWarning(), rcppExportWarning(), and Rcpp::attributes::trimWhitespace().

Referenced by parseAttribute().

Here is the call graph for this function:



6.724.3.14 std::vector< Param > Rcpp::attributes::SourceFileAttributesParser::parseParameters (const std::string & *input*) [private]

Definition at line 1343 of file attributes.cpp.

Referenced by parseAttribute().

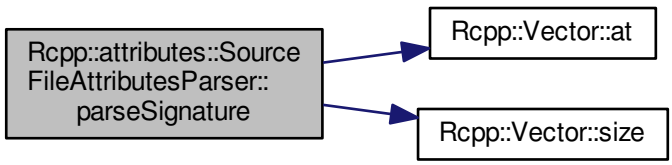
6.724.3.15 std::string Rcpp::attributes::SourceFileAttributesParser::parseSignature (size_t *lineNumber*) [private]

Definition at line 1494 of file attributes.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::at(), lines_, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by parseFunction().

Here is the call graph for this function:



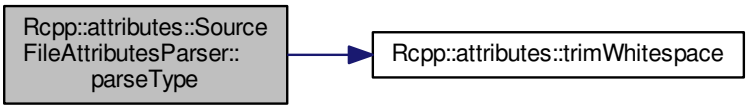
6.724.3.16 Type Rcpp::attributes::SourceFileAttributesParser::parseType (const std::string & *text*) [private]

Definition at line 1585 of file attributes.cpp.

References Rcpp::attributes::trimWhitespace().

Referenced by parseFunction().

Here is the call graph for this function:



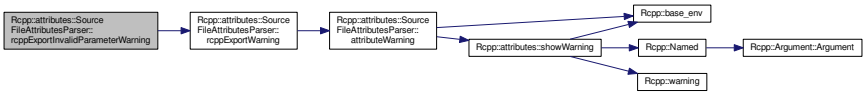
6.724.3.17 void Rcpp::attributes::SourceFileAttributesParser::rcppExportInvalidParameterWarning (const std::string & *param*, size_t *lineNumber*) [private]

Definition at line 1667 of file attributes.cpp.

References rcppExportWarning().

Referenced by parseFunction().

Here is the call graph for this function:



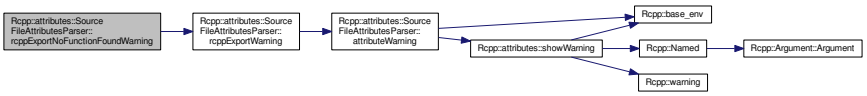
6.724.3.18 void Rcpp::attributes::SourceFileAttributesParser::rcppExportNoFunctionFoundWarning (size_t *lineNumber*) [private]

Definition at line 1662 of file attributes.cpp.

References rcppExportWarning().

Referenced by parseFunction().

Here is the call graph for this function:



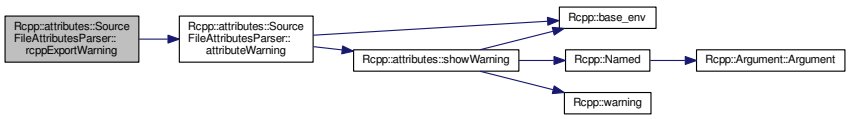
6.724.3.19 void Rcpp::attributes::SourceFileAttributesParser::rcppExportWarning (const std::string & *message,* size_t *lineNumber*) [private]

Definition at line 1656 of file attributes.cpp.

References attributeWarning().

Referenced by parseAttribute(), parseFunction(), rcppExportInvalidParameterWarning(), and rcppExportNoFunction↩FoundWarning().

Here is the call graph for this function:



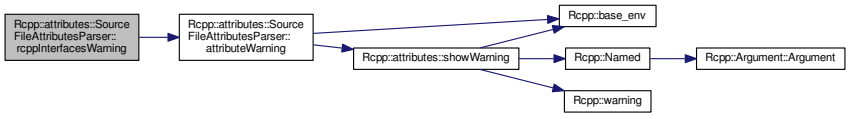
6.724.3.20 void Rcpp::attributes::SourceFileAttributesParser::rcppInterfacesWarning (const std::string & *message,* size_t *lineNumber*) [private]

Definition at line 1674 of file attributes.cpp.

References attributeWarning().

Referenced by parseAttribute().

Here is the call graph for this function:



6.724.3.21 virtual const std::vector<std::vector<std::string> >& Rcpp::attributes::SourceFileAttributesParser::roxygenChunks ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 468 of file attributes.cpp.

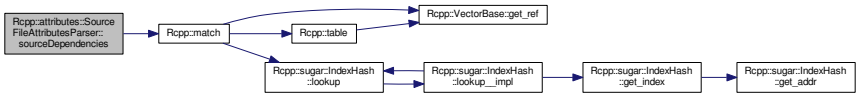
6.724.3.22 const std::vector<FileInfo>& Rcpp::attributes::SourceFileAttributesParser::sourceDependencies ( ) const [inline]

Definition at line 500 of file attributes.cpp.

References Rcpp::match().

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

Here is the call graph for this function:



6.724.3.23 virtual const std::string& Rcpp::attributes::SourceFileAttributesParser::sourceFile ( ) const [inline], [virtual]

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 457 of file attributes.cpp.

6.724.4 Member Data Documentation

6.724.4.1 std::vector<Attribute> Rcpp::attributes::SourceFileAttributesParser::attributes_ [private]

Definition at line 531 of file attributes.cpp.

Referenced by SourceFileAttributesParser().

6.724.4.2 std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::embeddedR_ [private]

Definition at line 533 of file attributes.cpp.

Referenced by SourceFileAttributesParser().

6.724.4.3 CharacterVector Rcpp::attributes::SourceFileAttributesParser::lines_ [private]

Definition at line 530 of file attributes.cpp.

Referenced by parseAttribute(), parseSignature(), and SourceFileAttributesParser().

6.724.4.4 std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::modules_ [private]

Definition at line 532 of file attributes.cpp.

Referenced by SourceFileAttributesParser().

6.724.4.5 std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::roxygenBuffer_ [private]

Definition at line 536 of file attributes.cpp.

Referenced by parseAttribute(), and SourceFileAttributesParser().

6.724.4.6 std::vector<std::vector<std::string> > Rcpp::attributes::SourceFileAttributesParser::roxygenChunks_ [private]

Definition at line 535 of file attributes.cpp.

Referenced by SourceFileAttributesParser().

6.724.4.7 std::vector<FileInfo> Rcpp::attributes::SourceFileAttributesParser::sourceDependencies_ [private]

Definition at line 534 of file attributes.cpp.

Referenced by SourceFileAttributesParser().

6.724.4.8 std::string Rcpp::attributes::SourceFileAttributesParser::sourceFile_ [private]

Definition at line 529 of file attributes.cpp.

Referenced by attributeWarning(), and SourceFileAttributesParser().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.725 Rcpp::algorithm::helpers::sqrt Struct Reference

`#include <algorithm.h>`

Public Member Functions

- template<typename T >
  double operator() (T val)

6.725.1 Detailed Description

Definition at line 198 of file algorithm.h.

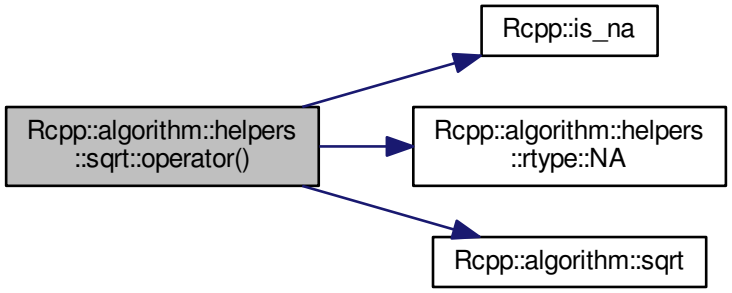
6.725.2 Member Function Documentation

6.725.2.1 template<typename T > double Rcpp::algorithm::helpers::sqrt::operator() (T *val*) [inline]

Definition at line 200 of file algorithm.h.

References Rcpp::is_na(), Rcpp::algorithm::helpers::rtype< T >::NA(), Rcpp::algorithm::sqrt(), and Rcpp::algorithm::helpers::ctype_helper< I >::value.

Here is the call graph for this function:

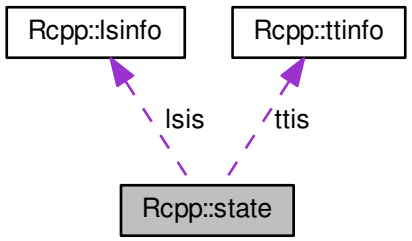


The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.726 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]

6.726.1 Detailed Description

Definition at line 334 of file Date.cpp.

6.726.2 Member Data Documentation

6.726.2.1 time_t Rcpp::state::ats[TZ_MAX_TIMES]

Definition at line 341 of file Date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.2 int Rcpp::state::charcnt

Definition at line 338 of file Date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.3 char Rcpp::state::chars[BIGGEST(BIGGEST(TZ_MAX_CHARS+1, sizeof gmt),(2 ∗(MY_TZNAME_MAX+1)))]

Definition at line 345 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.4 int Rcpp::state::goahead

Definition at line 340 of file Date.cpp.

Referenced by Rcpp::tzload().

6.726.2.5 int Rcpp::state::goback

Definition at line 339 of file Date.cpp.

Referenced by Rcpp::tzload().

6.726.2.6 int Rcpp::state::leapcnt

Definition at line 335 of file Date.cpp.

Referenced by Rcpp::timesub(), Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.7 struct lsinfo Rcpp::state::lsis[TZ_MAX_LEAPS]

Definition at line 346 of file Date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

6.726.2.8 int Rcpp::state::timecnt

Definition at line 336 of file Date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.9 struct ttinfo Rcpp::state::ttis[TZ_MAX_TYPES]

Definition at line 343 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.10 int Rcpp::state::typecnt

Definition at line 337 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.726.2.11 unsigned char Rcpp::state::types[TZ_MAX_TIMES]

Definition at line 342 of file Date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

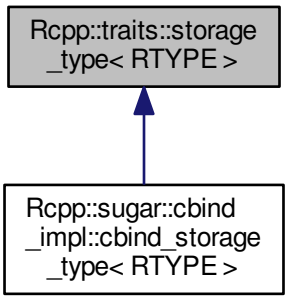
The documentation for this struct was generated from the following file:

- src/Date.cpp

6.727 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

6.727.1 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.

6.727.2 Member Typedef Documentation

6.727.2.1 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

6.728 Rcpp::traits::storage_type< CPLXSXP > Struct Template Reference

`#include <storage_type.h>`

Public Types

- typedef Rcomplex type

6.728.1 Detailed Description

template<>
struct Rcpp::traits::storage_type< CPLXSXP >

Total specialization for numeric vectors (CPLXSXP) typedef to Rcomplex

Definition at line 59 of file storage_type.h.

6.728.2 Member Typedef Documentation

6.728.2.1 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

6.729 Rcpp::traits::storage_type< INTSXP > Struct Template Reference

`#include <storage_type.h>`

Public Types

- typedef int type

6.729.1 Detailed Description

template<>
struct Rcpp::traits::storage_type< INTSXP >

Total specialization for integer vector (INTSXP) typedef to int

Definition at line 43 of file storage_type.h.

6.729.2 Member Typedef Documentation

6.729.2.1 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

6.730 Rcpp::traits::storage_type< LGLSXP > Struct Template Reference

`#include <storage_type.h>`

Public Types

- typedef int type

6.730.1 Detailed Description

template<>
struct Rcpp::traits::storage_type< LGLSXP >

Total specialization for logical vectors (LGLSXP) typedef to int

Definition at line 75 of file storage_type.h.

6.730.2 Member Typedef Documentation

6.730.2.1 typedef int Rcpp::traits::storage_type< LGLSXP >::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

6.731 Rcpp::traits::storage_type< RAWSXP > Struct Template Reference

`#include <storage_type.h>`

Public Types

- typedef Rbyte type

6.731.1 Detailed Description

template<>
struct Rcpp::traits::storage_type< RAWSXP >

Total specialization for raw vectors (RAWSXP) typedef to Rbyte

Definition at line 67 of file storage_type.h.

6.731.2 Member Typedef Documentation

6.731.2.1 typedef Rbyte Rcpp::traits::storage_type< RAWSXP >::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

6.732 Rcpp::traits::storage_type< REALSXP > Struct Template Reference

`#include <storage_type.h>`

Public Types

- typedef double type

6.732.1 Detailed Description

template<>
struct Rcpp::traits::storage_type< REALSXP >

Total specialization for numeric vectors (REALSXP) typedef to double

Definition at line 51 of file storage_type.h.

6.732.2 Member Typedef Documentation

6.732.2.1 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

6.733 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 &other)
- String (SEXP charsxp)
- 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 () 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
- std::string buffer
- bool valid
- bool buffer_ready
- cetype_t enc

6.733.1 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.

6.733.2 Member Typedef Documentation

6.733.2.1 typedef internal::const_string_proxy<STRSXP> Rcpp::String::const_StringProxy

Definition at line 52 of file String.h.

6.733.2.2 typedef internal::string_proxy<STRSXP> Rcpp::String::StringProxy

Definition at line 51 of file String.h.

6.733.3 Constructor & Destructor Documentation

6.733.3.1 Rcpp::String::String ( ) [inline]

default constructor

Definition at line 55 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Referenced by replace_all().

Here is the call graph for this function:

6.733.3.2 Rcpp::String::String (const String & *other*) [inline]

copy constructor

Definition at line 61 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Here is the call graph for this function:

6.733.3.3 Rcpp::String::String (SEXP *charsxp*) [inline]

construct a string from a single CHARSXP SEXP

Definition at line 67 of file String.h.

References buffer_ready, data, enc, Rcpp::Rcpp_PreserveObject(), RCPP_STRING_DEBUG, and valid.

Here is the call graph for this function:

6.733.3.4 Rcpp::String::String (const StringProxy & *proxy*) [inline]

from string proxy

Definition at line 85 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

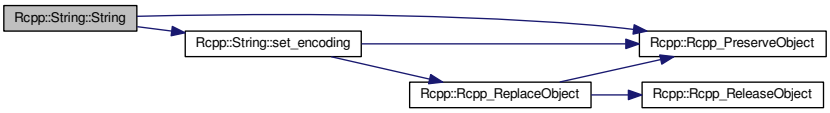
Here is the call graph for this function:

6.733.3.5 Rcpp::String::String (const StringProxy & *proxy,* cetype_t *enc*) [inline]

Definition at line 90 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), RCPP_STRING_DEBUG, and set_encoding().

Here is the call graph for this function:



6.733.3.6 Rcpp::String::String (const const_StringProxy & *proxy*) [inline]

from string proxy

Definition at line 97 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

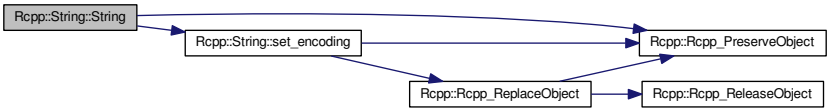
Here is the call graph for this function:

6.733.3.7 Rcpp::String::String (const const_StringProxy & *proxy*, cetype_t *enc*) [inline]

Definition at line 102 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), RCPP_STRING_DEBUG, and set_encoding().

Here is the call graph for this function:



6.733.3.8 Rcpp::String::String (const std::string & *s*, cetype_t *enc* = CE_UTF8) [inline]

from a std::string

Definition at line 109 of file String.h.

References data, and RCPP_STRING_DEBUG.

6.733.3.9 Rcpp::String::String (const std::wstring & *s*, cetype_t *enc* = CE_UTF8) [inline]

Definition at line 114 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Here is the call graph for this function:

6.733.3.10 Rcpp::String::String (const char * *s*, cetype_t *enc* = CE_UTF8) [inline]

from a const char*

Definition at line 120 of file String.h.

References data, and RCPP_STRING_DEBUG.

6.733.3.11 Rcpp::String::String (const wchar_t * *s*, cetype_t *enc* = CE_UTF8) [inline]

Definition at line 125 of file String.h.

References data, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Here is the call graph for this function:

6.733.3.12 Rcpp::String::String (int *x*) [inline]

constructors from R primitives

Definition at line 131 of file String.h.

References data, and Rcpp::Rcpp_PreserveObject().

Here is the call graph for this function:

6.733.3.13 Rcpp::String::String (double *x*) [inline]

Definition at line 132 of file String.h.

References data, and Rcpp::Rcpp_PreserveObject().

Here is the call graph for this function:

6.733.3.14 Rcpp::String::String (bool *x*) [inline]

Definition at line 133 of file String.h.

References data, and Rcpp::Rcpp_PreserveObject().

Here is the call graph for this function:

6.733.3.15 Rcpp::String::String (Rcomplex *x*) [inline]

Definition at line 134 of file String.h.

References data, and Rcpp::Rcpp_PreserveObject().

Here is the call graph for this function:

6.733.3.16 Rcpp::String::String (Rbyte *x*) [inline]

Definition at line 135 of file String.h.

References data, and Rcpp::Rcpp_PreserveObject().

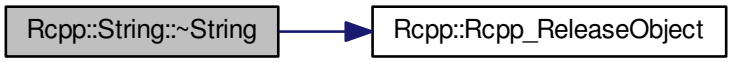
Here is the call graph for this function:

6.733.3.17 Rcpp::String::~String ( ) [inline]

Definition at line 137 of file String.h.

References data, and Rcpp::Rcpp_ReleaseObject().

Here is the call graph for this function:



6.733.4 Member Function Documentation

6.733.4.1 template<typename T > void Rcpp::String::append (const T & *s*) [inline], [private]

Definition at line 472 of file String.h.

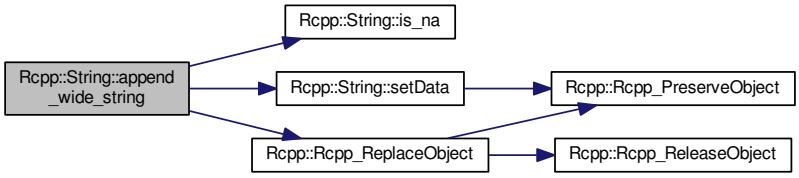
6.733.4.2 template<typename T > String& Rcpp::String::append_wide_string (const T & *s*) [inline], [private]

Definition at line 182 of file String.h.

References buffer_ready, data, DEMANGLE, is_na(), Rcpp::Rcpp_ReplaceObject(), RCPP_STRING_DEBUG_1, setData(), and valid.

Referenced by operator+=().

Here is the call graph for this function:



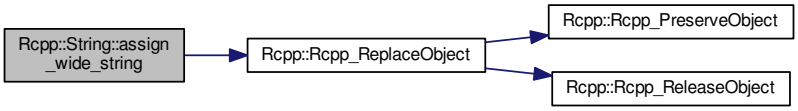
6.733.4.3 template<typename T > String& Rcpp::String::assign_wide_string (const T & *s*) [inline], [private]

Definition at line 156 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Referenced by operator=().

Here is the call graph for this function:



6.733.4.4 const char* Rcpp::String::get_cstring ( ) const [inline]

Definition at line 381 of file String.h.

References buffer, buffer_ready, and data.

Referenced by operator std::string(), operator std::wstring(), Rcpp::operator!=(), operator<(), operator>(), push_back(), push_front(), replace_all(), replace_first(), and replace_last().

6.733.4.5 cetype_t Rcpp::String::get_encoding ( ) const [inline]

Definition at line 385 of file String.h.

References enc.

6.733.4.6 SEXP Rcpp::String::get_sexp ( ) const [inline]

Definition at line 362 of file String.h.

References buffer, data, enc, RCPP_STRING_DEBUG_1, and valid.

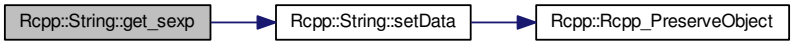
Referenced by Rcpp::internal::string_element_converter< RTYPE >::get(), operator!=(), Rcpp::internal::string_proxy< RTYPE >::operator=(), operator=(), operator==(), and Rcpp::internal::range_wrap_dispatch___impl__pair().

6.733.4.7 SEXP Rcpp::String::get_sexp ( ) [inline]

Definition at line 367 of file String.h.

References data, RCPP_STRING_DEBUG_1, setData(), and valid.

Here is the call graph for this function:



6.733.4.8 bool Rcpp::String::is_na ( ) const [inline], [private]

Definition at line 457 of file String.h.

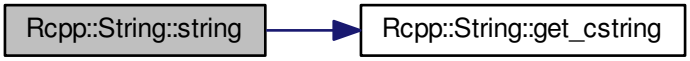
Referenced by append_wide_string(), operator+=(), push_back(), push_front(), replace_all(), replace_first(), and replace_last().

6.733.4.9 Rcpp::String::operator std::string ( ) const [inline]

Definition at line 372 of file String.h.

References get_cstring().

Here is the call graph for this function:

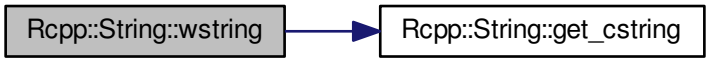


6.733.4.10 Rcpp::String::operator std::wstring ( ) const [inline]

Definition at line 376 of file String.h.

References get_cstring().

Here is the call graph for this function:



6.733.4.11 bool Rcpp::String::operator!= (const Rcpp::String & *other*) const [inline]

Definition at line 408 of file String.h.

References get_sexp().

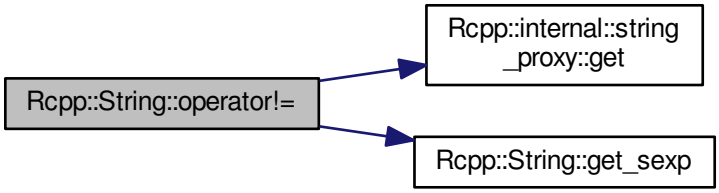
Here is the call graph for this function:

6.733.4.12 bool Rcpp::String::operator!= (const StringProxy & *other*) const [inline]

Definition at line 416 of file String.h.

References Rcpp::internal::string_proxy< RTYPE >::get(), and get_sexp().

Here is the call graph for this function:

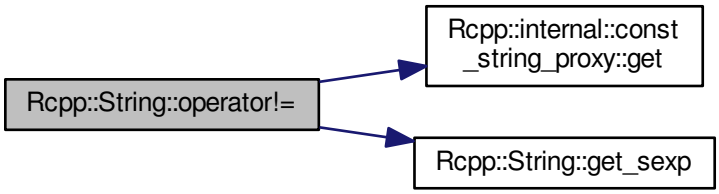


6.733.4.13 bool Rcpp::String::operator!= (const const_StringProxy & *other*) const [inline]

Definition at line 424 of file String.h.

References Rcpp::internal::const_string_proxy< RTYPE >::get(), and get_sexp().

Here is the call graph for this function:



6.733.4.14 bool Rcpp::String::operator!= (SEXP *other*) const [inline]

Definition at line 436 of file String.h.

References get_sexp().

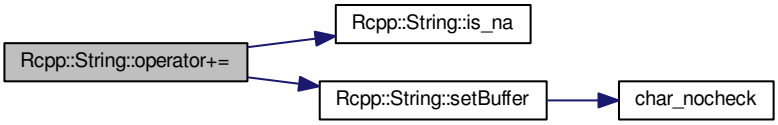
Here is the call graph for this function:

6.733.4.15 String& Rcpp::String::operator+= (const std::string & *s*) [inline]

Definition at line 167 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:

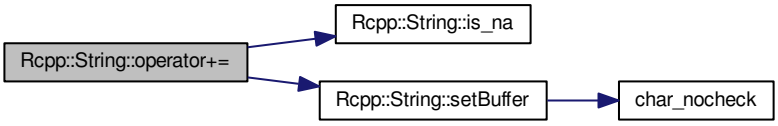


6.733.4.16 String& Rcpp::String::operator+= (const char * *s*) [inline]

Definition at line 174 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:

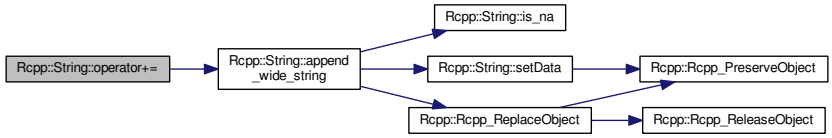


6.733.4.17 String& Rcpp::String::operator+= (const std::wstring & *s*) [inline]

Definition at line 197 of file String.h.

References append_wide_string().

Here is the call graph for this function:

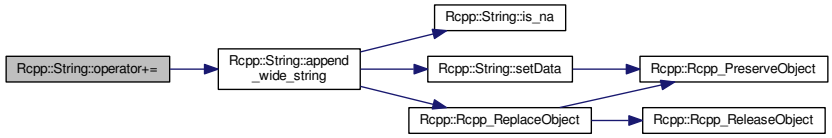


6.733.4.18 String& Rcpp::String::operator+= (const wchar_t * *s*) [inline]

Definition at line 198 of file String.h.

References append_wide_string().

Here is the call graph for this function:

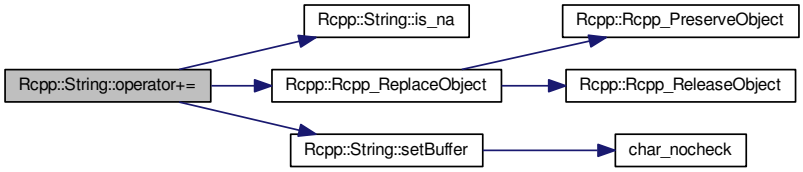


6.733.4.19 String& Rcpp::String::operator+= (const String & *other*) [inline]

Definition at line 200 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_ReplaceObject(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:

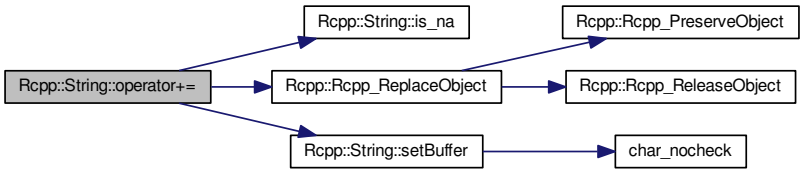


6.733.4.20 String& Rcpp::String::operator+= (const StringProxy & *proxy*) [inline]

Definition at line 207 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_ReplaceObject(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:

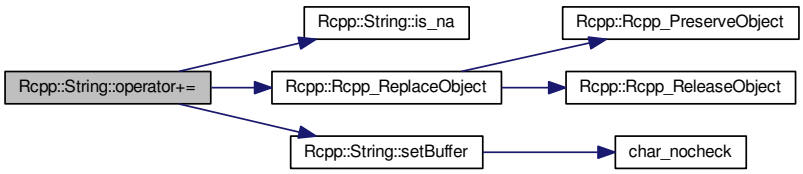


6.733.4.21 String& Rcpp::String::operator+= (const const_StringProxy & *proxy*) [inline]

Definition at line 215 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_ReplaceObject(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:

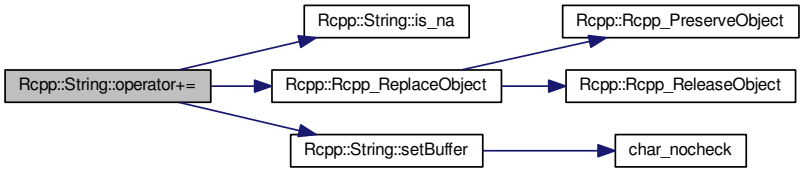


6.733.4.22 String& Rcpp::String::operator+= (SEXP *x*) [inline]

Definition at line 223 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_ReplaceObject(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:



6.733.4.23 bool Rcpp::String::operator< (const Rcpp::String & *other*) const [inline]

Definition at line 401 of file String.h.

References get_cstring().

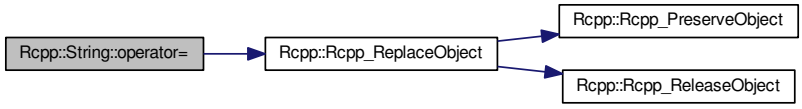
Here is the call graph for this function:

6.733.4.24 String& Rcpp::String::operator= (int *x*) [inline]

Definition at line 143 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:



6.733.4.25 String& Rcpp::String::operator= (double *x*) [inline]

Definition at line 144 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:

6.733.4.26 String& Rcpp::String::operator= (Rbyte *x*) [inline]

Definition at line 145 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:

6.733.4.27 String& Rcpp::String::operator= (bool *x*) [inline]

Definition at line 146 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:

6.733.4.28 String& Rcpp::String::operator= (Rcomplex *x*) [inline]

Definition at line 147 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:

6.733.4.29 String& Rcpp::String::operator= (SEXP *x*) [inline]

Definition at line 148 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

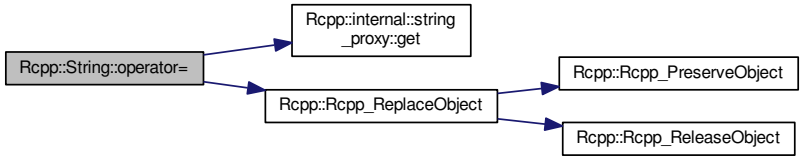
Here is the call graph for this function:

6.733.4.30 String& Rcpp::String::operator= (const StringProxy & *proxy*) [inline]

Definition at line 149 of file String.h.

References buffer_ready, data, Rcpp::internal::string_proxy< RTYPE >::get(), Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:

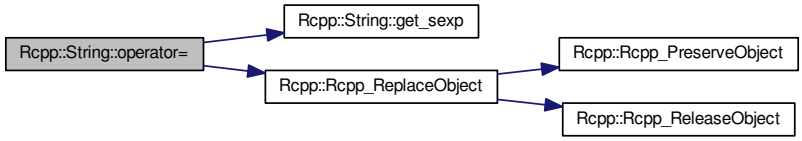


6.733.4.31 String& Rcpp::String::operator= (const String & *other*) [inline]

Definition at line 150 of file String.h.

References buffer_ready, data, get_sexp(), Rcpp::Rcpp_ReplaceObject(), and valid.

Here is the call graph for this function:



6.733.4.32 String& Rcpp::String::operator= (const std::string & *s*) [inline]

Definition at line 151 of file String.h.

References buffer, buffer_ready, and valid.

6.733.4.33 String& Rcpp::String::operator= (const char * *s*) [inline]

Definition at line 152 of file String.h.

References buffer, buffer_ready, and valid.

6.733.4.34 String& Rcpp::String::operator= (const std::wstring & *s*) [inline]

Definition at line 164 of file String.h.

References assign_wide_string().

Here is the call graph for this function:

6.733.4.35 String& Rcpp::String::operator= (const wchar_t * *s*) [inline]

Definition at line 165 of file String.h.

References assign_wide_string().

Here is the call graph for this function:

6.733.4.36 bool Rcpp::String::operator== (const Rcpp::String & *other*) const [inline]

Definition at line 405 of file String.h.

References get_sexp().

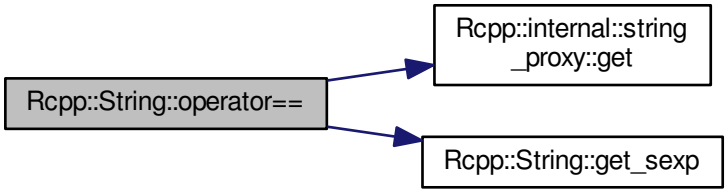
Here is the call graph for this function:

6.733.4.37 bool Rcpp::String::operator== (const StringProxy & *other*) const [inline]

Definition at line 412 of file String.h.

References Rcpp::internal::string_proxy< RTYPE >::get(), and get_sexp().

Here is the call graph for this function:

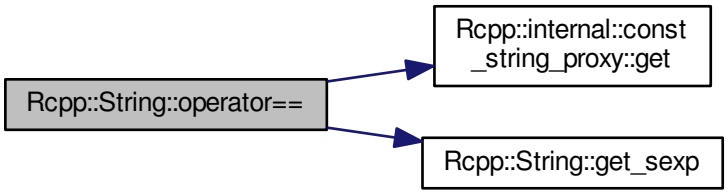


6.733.4.38 bool Rcpp::String::operator== (const const_StringProxy & *other*) const [inline]

Definition at line 420 of file String.h.

References Rcpp::internal::const_string_proxy< RTYPE >::get(), and get_sexp().

Here is the call graph for this function:



6.733.4.39 bool Rcpp::String::operator== (SEXP *other*) const [inline]

Definition at line 432 of file String.h.

References get_sexp().

Here is the call graph for this function:

6.733.4.40 bool Rcpp::String::operator> (const Rcpp::String & *other*) const [inline]

Definition at line 428 of file String.h.

References get_cstring().

Here is the call graph for this function:

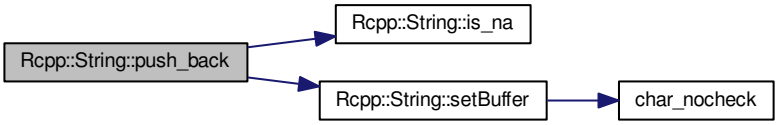
6.733.4.41 String& Rcpp::String::push_back (const char * *s*) [inline]

Definition at line 327 of file String.h.

References buffer, is_na(), setBuffer(), and valid.

Referenced by push_back().

Here is the call graph for this function:

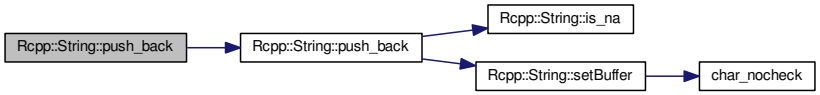


6.733.4.42 String& Rcpp::String::push_back (const std::string & *s*) [inline]

Definition at line 332 of file String.h.

References push_back().

Here is the call graph for this function:

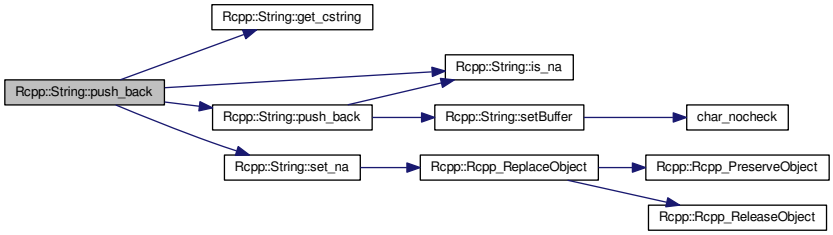


6.733.4.43 String& Rcpp::String::push_back (const Rcpp::String & *s*) [inline]

Definition at line 335 of file String.h.

References get_cstring(), is_na(), push_back(), and set_na().

Here is the call graph for this function:



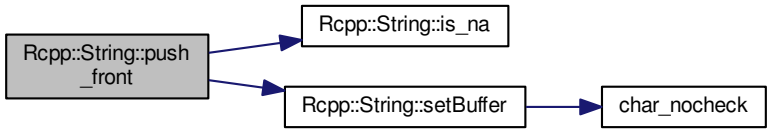
6.733.4.44 String& Rcpp::String::push_front (const char * *s*) [inline]

Definition at line 341 of file String.h.

References buffer, is_na(), setBuffer(), and valid.

Referenced by push_front().

Here is the call graph for this function:

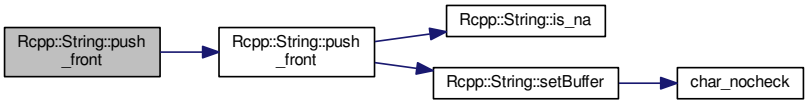


6.733.4.45 String& Rcpp::String::push_front (const std::string & *s*) [inline]

Definition at line 346 of file String.h.

References push_front().

Here is the call graph for this function:

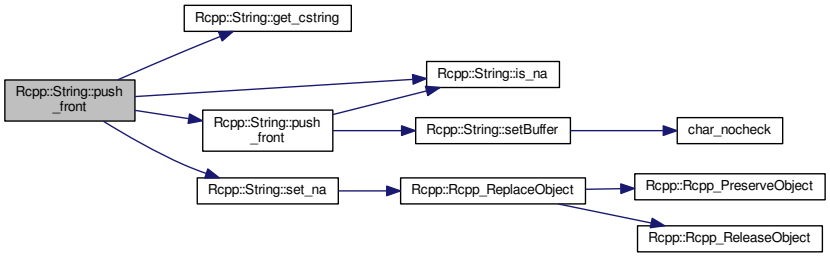


6.733.4.46 String& Rcpp::String::push_front (const Rcpp::String & *s*) [inline]

Definition at line 349 of file String.h.

References get_cstring(), is_na(), push_front(), and set_na().

Here is the call graph for this function:



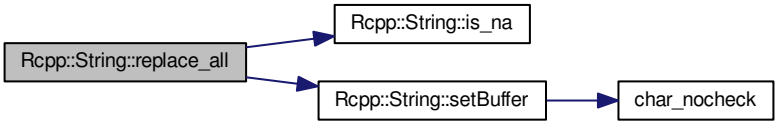
6.733.4.47 String& Rcpp::String::replace_all (const char * *s,* const char * *news*) [inline]

Definition at line 290 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:

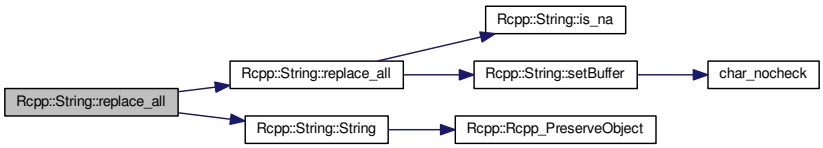


6.733.4.48 template<typename LHS , typename RHS > String& Rcpp::String::replace_all (const LHS & *s,* const RHS & *news*) [inline]

Definition at line 307 of file String.h.

References replace_all(), and String().

Here is the call graph for this function:

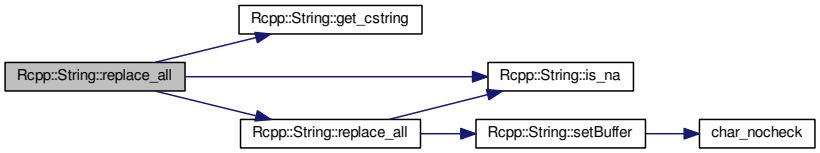


6.733.4.49 String& Rcpp::String::replace_all (const Rcpp::String & *s,* const char * *news*) [inline]

Definition at line 311 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:

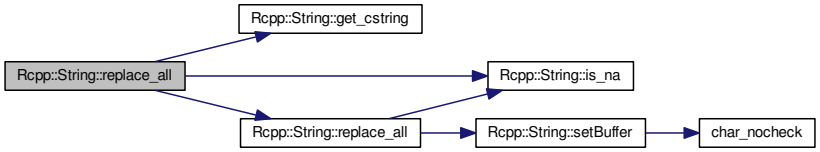


6.733.4.50 String& Rcpp::String::replace_all (const char * *s*, const Rcpp::String & *news*) [inline]

Definition at line 316 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:

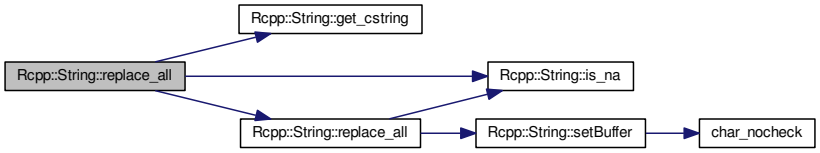


6.733.4.51 String& Rcpp::String::replace_all (const Rcpp::String & *s*, const Rcpp::String & *news*) [inline]

Definition at line 321 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:



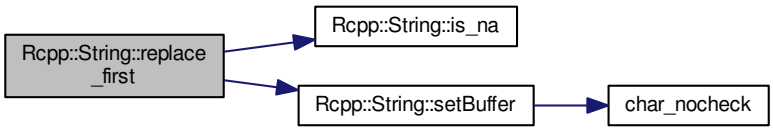
6.733.4.52 String& Rcpp::String::replace_first (const char * *s*, const char * *news*) [inline]

Definition at line 237 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:

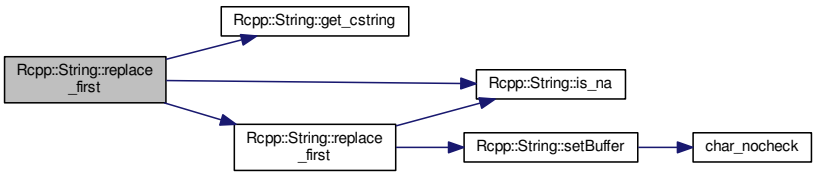


6.733.4.53 String& Rcpp::String::replace_first (const Rcpp::String & *s*, const char * *news*) [inline]

Definition at line 247 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:

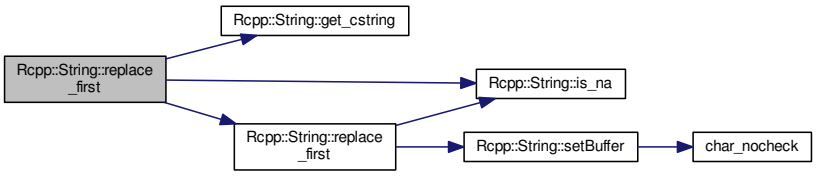


6.733.4.54 String& Rcpp::String::replace_first (const char * *s*, const Rcpp::String & *news*) [inline]

Definition at line 252 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:

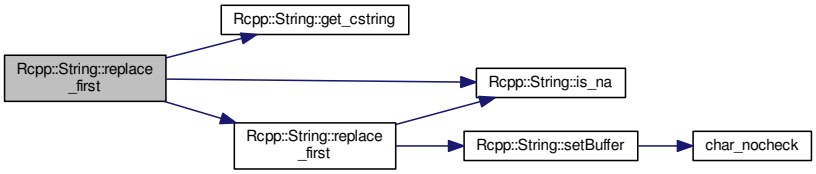


6.733.4.55 String& Rcpp::String::replace_first (const Rcpp::String & *s*, const Rcpp::String & *news*) [inline]

Definition at line 257 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:



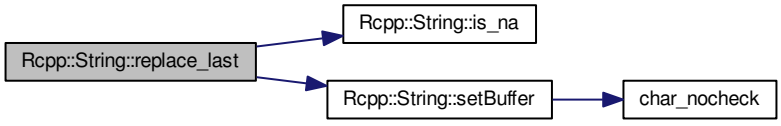
6.733.4.56 String& Rcpp::String::replace_last (const char * *s*, const char * *news*) [inline]

Definition at line 263 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:

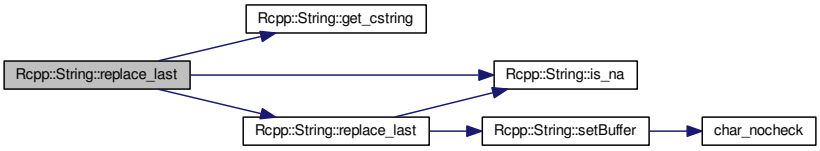


6.733.4.57 String& Rcpp::String::replace_last (const Rcpp::String & *s*, const char * *news*) [inline]

Definition at line 273 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:

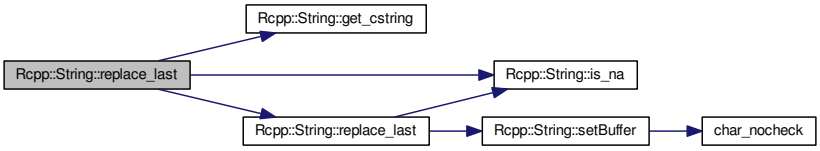


6.733.4.58 String& Rcpp::String::replace_last (const char * *s*, const Rcpp::String & *news*) [inline]

Definition at line 278 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:

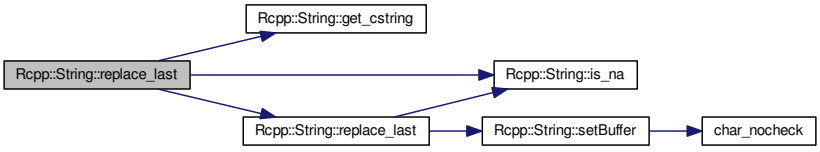


6.733.4.59 String& Rcpp::String::replace_last (const Rcpp::String & *s*, const Rcpp::String & *news*) [inline]

Definition at line 283 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:



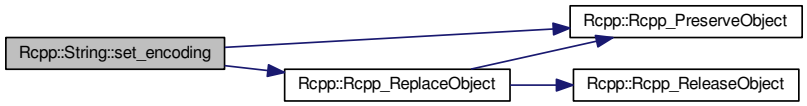
6.733.4.60 void Rcpp::String::set_encoding (cetype_t *encoding*) [inline]

Definition at line 389 of file String.h.

References buffer, data, enc, Rcpp::Rcpp_PreserveObject(), Rcpp::Rcpp_ReplaceObject(), and valid.

Referenced by String().

Here is the call graph for this function:



6.733.4.61 void Rcpp::String::set_na ( ) [inline]

Definition at line 356 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_ReplaceObject(), and valid.

Referenced by push_back(), and push_front().

Here is the call graph for this function:

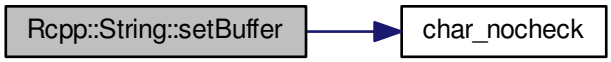
6.733.4.62 void Rcpp::String::setBuffer ( ) [inline], [private]

Definition at line 458 of file String.h.

References char_nocheck().

Referenced by operator+=(), push_back(), push_front(), replace_all(), replace_first(), and replace_last().

Here is the call graph for this function:



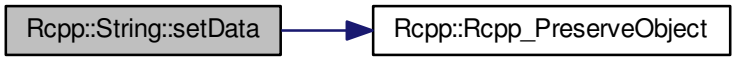
6.733.4.63 void Rcpp::String::setData ( ) [inline], [private]

Definition at line 464 of file String.h.

References enc, Rcpp::Rcpp_PreserveObject(), and RCPP_STRING_DEBUG.

Referenced by append_wide_string(), and get_sexp().

Here is the call graph for this function:



6.733.5 Member Data Documentation

6.733.5.1 std::string Rcpp::String::buffer [private]

a buffer used to do string operations withough going back to the SEXP

Definition at line 446 of file String.h.

Referenced by get_cstring(), get_sexp(), operator+=(), operator=(), push_back(), push_front(), replace_all(), replace↩ _first(), replace_last(), and set_encoding().

6.733.5.2 bool Rcpp::String::buffer_ready [private]

is the buffer initialized

Definition at line 452 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_cstring(), operator+=(), operator=(), set_na(), and String().

6.733.5.3 SEXP Rcpp::String::data [private]

the CHARSXP this String encapsulates

Definition at line 443 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_cstring(), get_sexp(), operator+=(), operator=(), set_encoding(), set_na(), String(), Rcpp::wrap< Rcpp::String >(), and ~String().

6.733.5.4 cetype_t Rcpp::String::enc [private]

the encoding of encapsulated CHARSXP

Definition at line 455 of file String.h.

Referenced by get_encoding(), get_sexp(), set_encoding(), setData(), and String().

6.733.5.5 bool Rcpp::String::valid [private]

is data in sync with buffer

Definition at line 449 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_sexp(), operator+=(), operator=(), push_back(), push_front(), replace_all(), replace_first(), replace_last(), set_encoding(), set_na(), and String().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/String.h

6.734 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)

6.734.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::string_element_converter< RTYPE >

Definition at line 43 of file converter.h.

6.734.2 Member Typedef Documentation

6.734.2.1 template<int RTYPE> typedef SEXP Rcpp::internal::string_element_converter< RTYPE >::target

Definition at line 45 of file converter.h.

6.734.3 Member Function Documentation

6.734.3.1 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.

6.734.3.2 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.

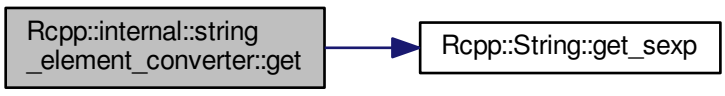
References RCPP_DEBUG.

6.734.3.3 template<int RTYPE> SEXP Rcpp::internal::string_element_converter< RTYPE >::get (const Rcpp::String & *input*) [static]

Definition at line 488 of file String.h.

References Rcpp::String::get_sexp(), and RCPP_DEBUG.

Here is the call graph for this function:



6.734.3.4 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.

6.734.3.5 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.

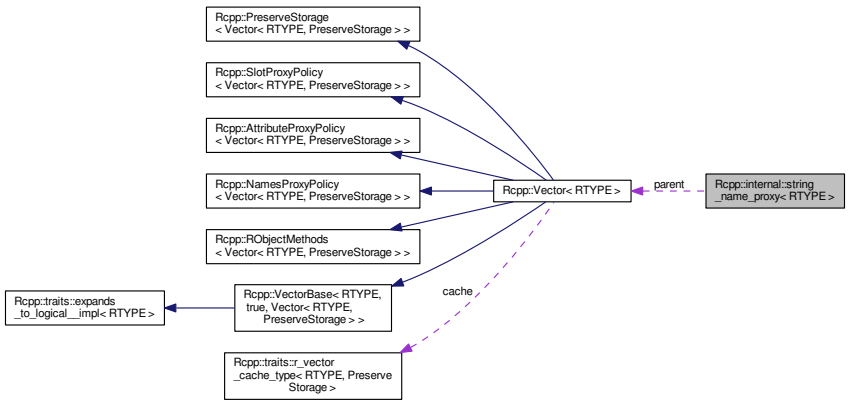
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/converter.h
- inst/include/Rcpp/String.h

6.735 Rcpp::internal::string_name_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Collaboration diagram for Rcpp::internal::string_name_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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 * ()
- operator SEXP ()
- 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 ()

Private Attributes

- VECTOR & parent
- std::string name

6.735.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::string_name_proxy< RTYPE >

Definition at line 33 of file 00_forward_proxy.h.

6.735.2 Member Typedef Documentation

6.735.2.1 template<int RTYPE> typedef const char* Rcpp::internal::string_name_proxy< RTYPE >::iterator

Definition at line 85 of file proxy.h.

6.735.2.2 template<int RTYPE> typedef const char& Rcpp::internal::string_name_proxy< RTYPE >::reference

Definition at line 86 of file proxy.h.

6.735.2.3 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::VECTOR

Definition at line 84 of file proxy.h.

6.735.3 Constructor & Destructor Documentation

6.735.3.1 template<int RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::string_name_proxy (VECTOR & *v*, const std::string & *name_*) [inline]

Definition at line 88 of file proxy.h.

6.735.3.2 template<int RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::string_name_proxy (const string_name_proxy< RTYPE > & *other*) [inline]

Definition at line 90 of file proxy.h.

6.735.3.3 template<int RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::~string_name_proxy ( ) [inline]

Definition at line 92 of file proxy.h.

6.735.4 Member Function Documentation

6.735.4.1 template<int RTYPE> iterator Rcpp::internal::string_name_proxy< RTYPE >::begin ( ) [inline]

Definition at line 111 of file proxy.h.

6.735.4.2 template<int RTYPE> iterator Rcpp::internal::string_name_proxy< RTYPE >::end ( ) [inline]

Definition at line 112 of file proxy.h.

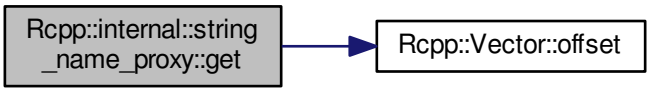
6.735.4.3 template<int RTYPE> char* Rcpp::internal::string_name_proxy< RTYPE >::get ( ) [inline], [private]

Definition at line 128 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Referenced by Rcpp::internal::string_name_proxy< RTYPE >::operator=().

Here is the call graph for this function:



6.735.4.4 template<int RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::operator char * ( ) [inline]

Definition at line 103 of file proxy.h.

6.735.4.5 template<int RTYPE> Rcpp::internal::string_name_proxy< RTYPE >::operator SEXP ( ) [inline]

Definition at line 107 of file proxy.h.

6.735.4.6 template<int RTYPE> string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE >::operator= (const std::string & *rhs*) [inline]

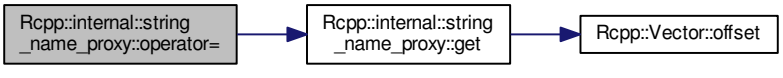
Definition at line 94 of file proxy.h.

6.735.4.7 template<int RTYPE> string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE >::operator= (const string_name_proxy< RTYPE > & *other*) [inline]

Definition at line 98 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE >::get().

Here is the call graph for this function:



6.735.4.8 template<int RTYPE> reference Rcpp::internal::string_name_proxy< RTYPE >::operator[ ] (R_xlen_t *i*) [inline]

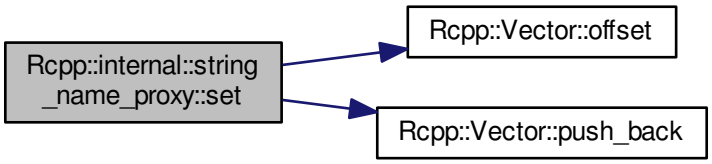
Definition at line 113 of file proxy.h.

6.735.4.9 template<int RTYPE> void Rcpp::internal::string_name_proxy< RTYPE >::set (const std::string & *rhs*) [inline], [private]

Definition at line 119 of file proxy.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Here is the call graph for this function:



6.735.4.10 template<int RTYPE> R_xlen_t Rcpp::internal::string_name_proxy< RTYPE >::size ( ) [inline]

Definition at line 114 of file proxy.h.

6.735.5 Member Data Documentation

6.735.5.1 template<int RTYPE> std::string Rcpp::internal::string_name_proxy< RTYPE >::name [private]

Definition at line 118 of file proxy.h.

6.735.5.2 template<int RTYPE> VECTOR& Rcpp::internal::string_name_proxy< RTYPE >::parent [private]

Definition at line 117 of file proxy.h.

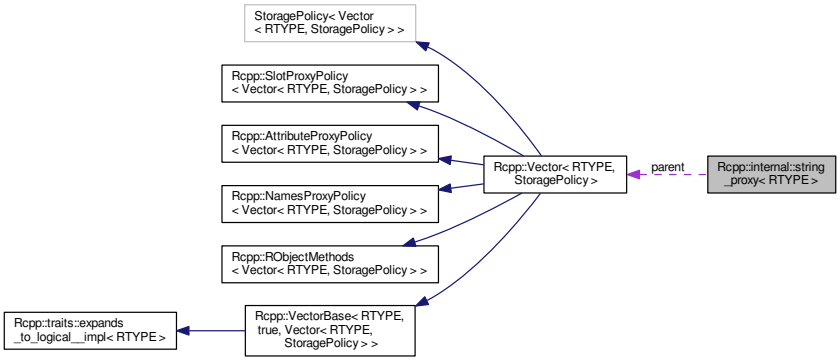
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.736 Rcpp::internal::string_proxy< RTYPE > Class Template Reference

`#include <00_forward_proxy.h>`

Collaboration diagram for Rcpp::internal::string_proxy< RTYPE >:



Public Types

- typedef ::Rcpp::Vector< RTYPE > 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 &other)
- string_proxy & operator= (const const_string_proxy< RTYPE > &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)
- bool operator!= (const char *other)
- bool operator== (const string_proxy &other)
- bool operator!= (const string_proxy &other)
- bool operator== (SEXP other) const
- bool operator!= (SEXP other) const
- template<typename T >
  string_proxy< RTYPE > & operator+= (const T &rhs)

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)

6.736.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::string_proxy< RTYPE >

Definition at line 28 of file 00_forward_proxy.h.

6.736.2 Member Typedef Documentation

6.736.2.1 template<int RTYPE> typedef const char* Rcpp::internal::string_proxy< RTYPE >::iterator

Definition at line 32 of file string_proxy.h.

6.736.2.2 template<int RTYPE> typedef const char& Rcpp::internal::string_proxy< RTYPE >::reference

Definition at line 33 of file string_proxy.h.

6.736.2.3 template<int RTYPE> typedef ::Rcpp::Vector<RTYPE> Rcpp::internal::string_proxy< RTYPE >::VECTOR

Definition at line 31 of file string_proxy.h.

6.736.3 Constructor & Destructor Documentation

6.736.3.1 template<int RTYPE> Rcpp::internal::string_proxy< RTYPE >::string_proxy ( ) [inline]

Definition at line 35 of file string_proxy.h.

6.736.3.2 template<int RTYPE> Rcpp::internal::string_proxy< RTYPE >::string_proxy (VECTOR & *v*, R_xlen_t *index_*) [inline]

Creates a proxy

Parameters

v	reference to the associated character vector
index	index

Definition at line 43 of file string_proxy.h.

6.736.3.3 template<int RTYPE> Rcpp::internal::string_proxy< RTYPE >::string_proxy (const string_proxy< RTYPE > & *other*) [inline]

Definition at line 45 of file string_proxy.h.

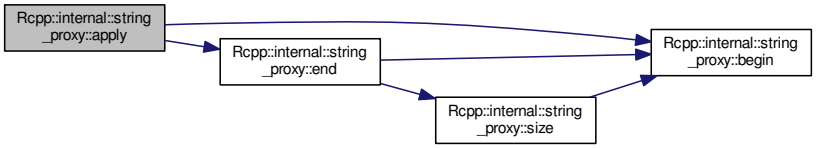
6.736.4 Member Function Documentation

6.736.4.1 template<int RTYPE> template<typename OutputIterator , typename UnaryOperator > void Rcpp::internal::string_proxy< RTYPE >::apply (OutputIterator *target,* UnaryOperator *op*) [inline]

Definition at line 172 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin(), and Rcpp::internal::string_proxy< RTYPE >::end().

Here is the call graph for this function:

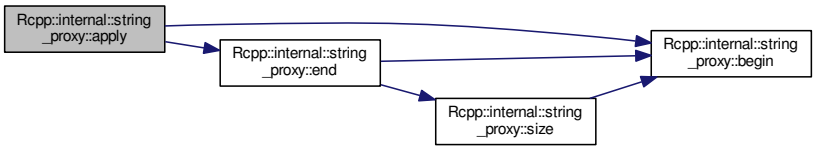


6.736.4.2 template<int RTYPE> template<typename UnaryOperator > void Rcpp::internal::string_proxy< RTYPE >::apply (UnaryOperator *op*) [inline]

Definition at line 177 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin(), and Rcpp::internal::string_proxy< RTYPE >::end().

Here is the call graph for this function:



6.736.4.3 template<int RTYPE> iterator Rcpp::internal::string_proxy< RTYPE >::begin ( ) const [inline]

Definition at line 158 of file string_proxy.h.

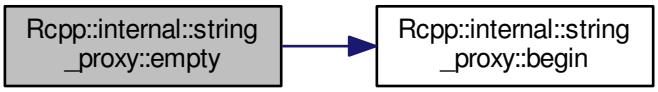
Referenced by Rcpp::internal::string_proxy< RTYPE >::apply(), Rcpp::internal::string_proxy< RTYPE >::empty(), Rcpp::internal::string_proxy< RTYPE >::end(), Rcpp::internal::string_proxy< RTYPE >::operator!=(), Rcpp::internal::string_proxy< RTYPE >::operator==(), Rcpp::internal::string_proxy< RTYPE >::operator[]( ), Rcpp::internal::string_proxy< RTYPE >::size(), and Rcpp::internal::string_proxy< RTYPE >::transform().

6.736.4.4 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::empty ( ) const [inline]

Definition at line 161 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:



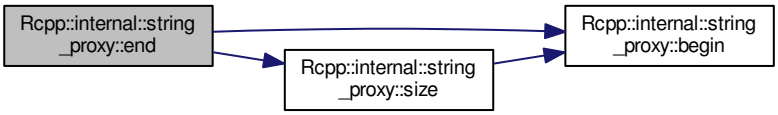
6.736.4.5 template<int RTYPE> iterator Rcpp::internal::string_proxy< RTYPE >::end ( ) const [inline]

Definition at line 159 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin(), and Rcpp::internal::string_proxy< RTYPE >::size().

Referenced by Rcpp::internal::string_proxy< RTYPE >::apply().

Here is the call graph for this function:



6.736.4.6 template<int RTYPE> SEXP Rcpp::internal::string_proxy< RTYPE >::get ( ) const [inline]

Definition at line 147 of file string_proxy.h.

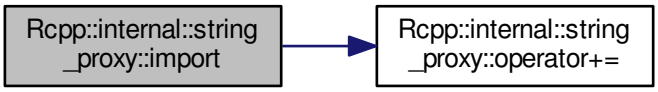
Referenced by Rcpp::String::operator!=(), Rcpp::String::operator=(), and Rcpp::String::operator==().

6.736.4.7 template<int RTYPE> void Rcpp::internal::string_proxy< RTYPE >::import (const string_proxy< RTYPE > & *other*) [inline]

Definition at line 93 of file string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE >::index, Rcpp::internal::string_proxy< RTYPE >::index, Rcpp::internal::string_proxy< RTYPE >::operator+=(), Rcpp::internal::const_string_proxy< RTYPE >::parent, and Rcpp::internal::string_proxy< RTYPE >::parent.

Here is the call graph for this function:



6.736.4.8 template<int RTYPE> void Rcpp::internal::string_proxy< RTYPE >::move (R_xlen_t *n*) [inline]

Definition at line 145 of file string_proxy.h.

6.736.4.9 template<int RTYPE> Rcpp::internal::string_proxy< RTYPE >::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 118 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::operator+.

6.736.4.10 template<int RTYPE> Rcpp::internal::string_proxy< RTYPE >::operator SEXP ( ) const [inline]

rhs use. Retrieves the current value of the element this proxy refers to.

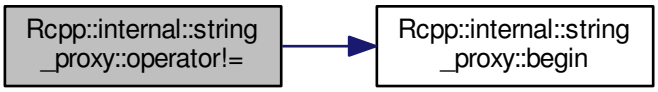
Definition at line 109 of file string_proxy.h.

6.736.4.11 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator!= (const char * *other*) [inline]

Definition at line 184 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:

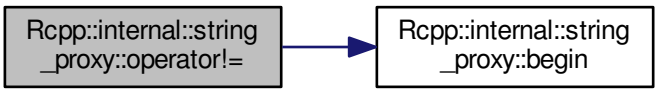


6.736.4.12 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator!= (const string_proxy< RTYPE > & *other*) [inline]

Definition at line 191 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:



6.736.4.13 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator!= (SEXP *other*) const [inline]

Definition at line 199 of file string_proxy.h.

6.736.4.14 template<int RTYPE> template<typename T > string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator+= (const T & *rhs*)

lhs use. Adds the content of the rhs proxy to the element this proxy refers to.

Referenced by Rcpp::internal::string_proxy< RTYPE >::import().

6.736.4.15 template<int RTYPE> template<typename T > string_proxy<RTYPE>& Rcpp::internal::string_proxy< RTYPE >::operator+= (const T & *rhs*)

Definition at line 500 of file String.h.

6.736.4.16 template<int RTYPE> string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator= (const string_proxy< RTYPE > & *other*) [inline]

lhs use. Assign the value of the referred element to the current content of the element referred by the rhs proxy

Parameters

rhs	another proxy, possibly from another vector

Definition at line 55 of file string_proxy.h.

6.736.4.17 template<int RTYPE> string_proxy< RTYPE > & Rcpp::internal::string_proxy< RTYPE >::operator= (const const_string_proxy< RTYPE > & *other*)

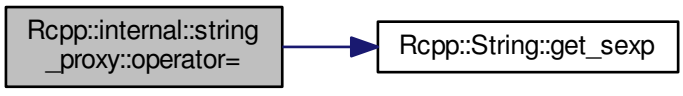
Definition at line 171 of file const_string_proxy.h.

6.736.4.18 template<int RTYPE> string_proxy< RTYPE > & Rcpp::internal::string_proxy< RTYPE >::operator= (const String & *s*)

Definition at line 482 of file String.h.

References Rcpp::String::get_sexp().

Here is the call graph for this function:



6.736.4.19 template<int RTYPE> template<typename T > string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator= (const std::basic_string< T > & *rhs*) [inline]

lhs use. Assigns the value of the referred element of the character vector

Parameters

rhs	new content for the element referred by this proxy

Definition at line 71 of file string_proxy.h.

6.736.4.20 template<int RTYPE> string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator= (const char * *rhs*) [inline]

Definition at line 76 of file string_proxy.h.

6.736.4.21 template<int RTYPE> string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator= (const wchar_t * *rhs*) [inline]

Definition at line 81 of file string_proxy.h.

6.736.4.22 template<int RTYPE> string_proxy& Rcpp::internal::string_proxy< RTYPE >::operator= (SEXP *rhs*) [inline]

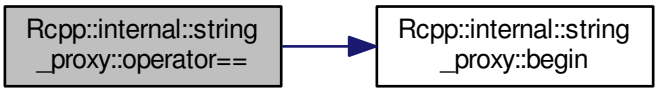
Definition at line 87 of file string_proxy.h.

6.736.4.23 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator== (const char * *other*) [inline]

Definition at line 181 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:

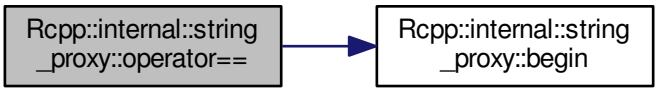


6.736.4.24 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator== (const string_proxy< RTYPE > & *other*) [inline]

Definition at line 188 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:



6.736.4.25 template<int RTYPE> bool Rcpp::internal::string_proxy< RTYPE >::operator== (SEXP *other*) const [inline]

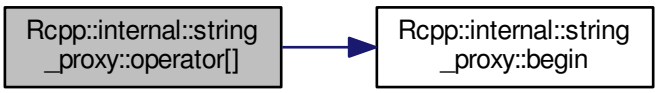
Definition at line 195 of file string_proxy.h.

6.736.4.26 template<int RTYPE> reference Rcpp::internal::string_proxy< RTYPE >::operator[] (R_xlen_t *n*) [inline]

Definition at line 162 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Here is the call graph for this function:



6.736.4.27 template<int RTYPE> template<typename T > void Rcpp::internal::string_proxy< RTYPE >::set (const T & *x*) [inline]

Definition at line 151 of file string_proxy.h.

6.736.4.28 template<int RTYPE> void Rcpp::internal::string_proxy< RTYPE >::set (SEXP *x*) [inline]

Definition at line 154 of file string_proxy.h.

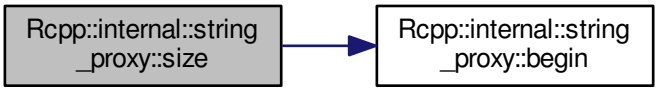
6.736.4.29 template<int RTYPE> R_xlen_t Rcpp::internal::string_proxy< RTYPE >::size ( ) const [inline]

Definition at line 160 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin().

Referenced by Rcpp::internal::string_proxy< RTYPE >::end().

Here is the call graph for this function:



6.736.4.30 template<int RTYPE> void Rcpp::internal::string_proxy< RTYPE >::swap (string_proxy< RTYPE > & *other*) [inline]

Definition at line 137 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::index, and Rcpp::internal::string_proxy< RTYPE >::parent.

6.736.4.31 template<int RTYPE> template<typename UnaryOperator > void Rcpp::internal::string_proxy< RTYPE >::transform (UnaryOperator *op*) [inline]

Definition at line 165 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE >::begin(), and Rcpp::internal::string_proxy< RTYPE >::buffer.

Here is the call graph for this function:

6.736.5 Friends And Related Function Documentation

6.736.5.1 template<int RTYPE> template<int RT> std::string operator+ (const std::string & *x*, const string_proxy< RT > & *proxy*) [friend]

Referenced by Rcpp::internal::string_proxy< RTYPE >::operator char *().

6.736.5.2 template<int RTYPE> 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

6.736.6 Member Data Documentation

6.736.6.1 template<int RTYPE> std::string Rcpp::internal::string_proxy< RTYPE >::buffer [static], [private]

Definition at line 205 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE >::transform().

6.736.6.2 template<int RTYPE> R_xlen_t Rcpp::internal::string_proxy< RTYPE >::index

Definition at line 144 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE >::import(), and Rcpp::internal::string_proxy< RTYPE >::swap().

6.736.6.3 template<int RTYPE> VECTOR* Rcpp::internal::string_proxy< RTYPE >::parent

Definition at line 143 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE >::import(), and Rcpp::internal::string_proxy< RTYPE >::swap().

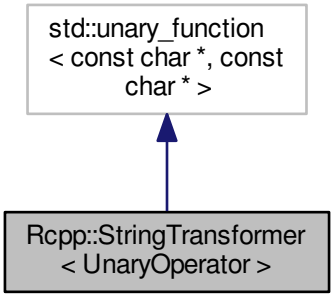
The documentation for this class was generated from the following files:

- 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

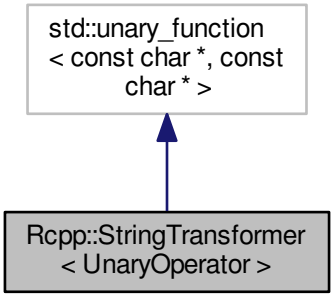
6.737 Rcpp::StringTransformer< UnaryOperator > Class Template Reference

`#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

6.737.1 Detailed Description

template<typename UnaryOperator>
class Rcpp::StringTransformer< UnaryOperator >

Definition at line 30 of file StringTransformer.h.

6.737.2 Constructor & Destructor Documentation

6.737.2.1 template<typename UnaryOperator > Rcpp::StringTransformer< UnaryOperator >::StringTransformer (const UnaryOperator & *op_*) [inline]

Definition at line 32 of file StringTransformer.h.

6.737.2.2 template<typename UnaryOperator > Rcpp::StringTransformer< UnaryOperator >::~StringTransformer ( ) [inline]

Definition at line 33 of file StringTransformer.h.

6.737.3 Member Function Documentation

6.737.3.1 template<typename UnaryOperator > const char* Rcpp::StringTransformer< UnaryOperator >::operator() (const char * *input*) [inline]

Definition at line 35 of file StringTransformer.h.

References Rcpp::StringTransformer< UnaryOperator >::buffer, and Rcpp::StringTransformer< UnaryOperator >::op.

6.737.4 Member Data Documentation

6.737.4.1 template<typename UnaryOperator > std::string Rcpp::StringTransformer< UnaryOperator >::buffer [private]

Definition at line 43 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

6.737.4.2 template<typename UnaryOperator > const UnaryOperator& Rcpp::StringTransformer< UnaryOperator >::op [private]

Definition at line 42 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

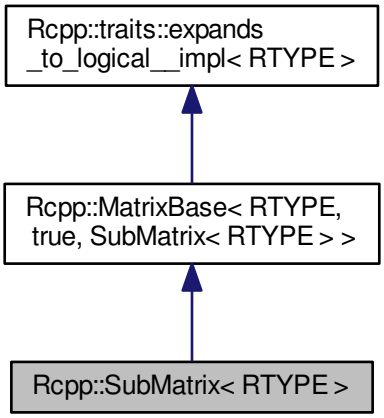
The documentation for this class was generated from the following file:

- inst/include/Rcpp/StringTransformer.h

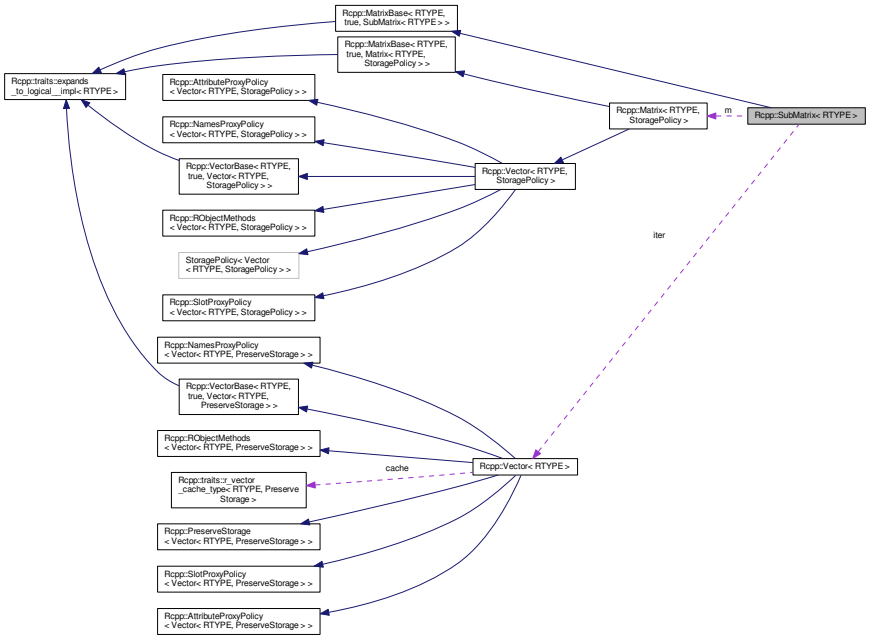
6.738 Rcpp::SubMatrix< RTYPE > Class Template Reference

`#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

6.738.1 Detailed Description

template<int RTYPE>
class Rcpp::SubMatrix< RTYPE >

Definition at line 28 of file SubMatrix.h.

6.738.2 Member Typedef Documentation

6.738.2.1 template<int RTYPE> typedef Matrix<RTYPE> Rcpp::SubMatrix< RTYPE >::MATRIX

Definition at line 30 of file SubMatrix.h.

6.738.2.2 template<int RTYPE> typedef MATRIX::Proxy Rcpp::SubMatrix< RTYPE >::Proxy

Definition at line 32 of file SubMatrix.h.

6.738.2.3 template<int RTYPE> typedef Vector<RTYPE>::iterator Rcpp::SubMatrix< RTYPE >::vec_iterator

Definition at line 31 of file SubMatrix.h.

6.738.3 Constructor & Destructor Documentation

6.738.3.1 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.

6.738.4 Member Function Documentation

6.738.4.1 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=().

6.738.4.2 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().

6.738.4.3 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().

6.738.4.4 template<int RTYPE> Proxy Rcpp::SubMatrix< RTYPE >::operator() (int *i*, int *j*) const [inline]

Definition at line 46 of file SubMatrix.h.

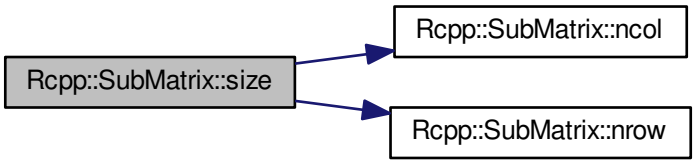
References Rcpp::SubMatrix< RTYPE >::iter, and Rcpp::SubMatrix< RTYPE >::m_nr.

6.738.4.5 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:



6.738.5 Member Data Documentation

6.738.5.1 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()().

6.738.5.2 template<int RTYPE> MATRIX& Rcpp::SubMatrix< RTYPE >::m [private]

Definition at line 53 of file SubMatrix.h.

6.738.5.3 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()().

6.738.5.4 template<int RTYPE> int Rcpp::SubMatrix< RTYPE >::nc [private]

Definition at line 55 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::ncol().

6.738.5.5 template<int RTYPE> int Rcpp::SubMatrix< RTYPE >::nr [private]

Definition at line 55 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::nrow().

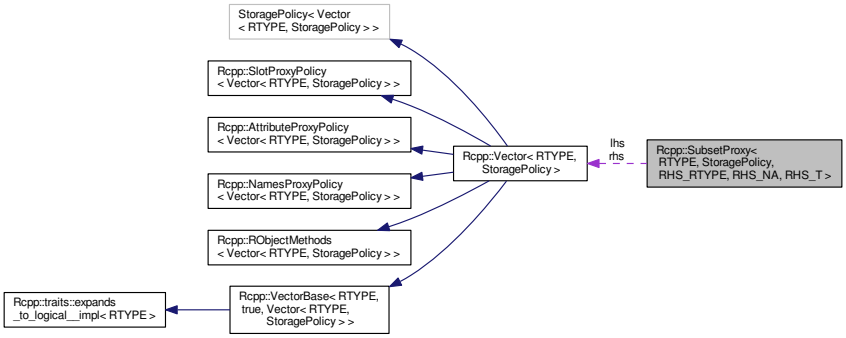
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/SubMatrix.h

6.739 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

- void check_indices (int *x, int n, int 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
- int lhs_n
- int rhs_n
- std::vector< int > indices
- int indices_n

6.739.1 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 31 of file Subsetter.h.

6.739.2 Member Typedef Documentation

6.739.2.1 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 33 of file Subsetter.h.

6.739.2.2 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 34 of file Subsetter.h.

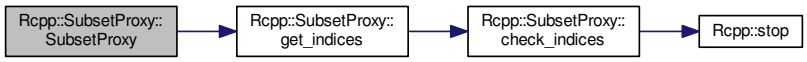
6.739.3 Constructor & Destructor Documentation

6.739.3.1 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 38 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:



6.739.3.2 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 43 of file Subsetter.h.

6.739.4 Member Function Documentation

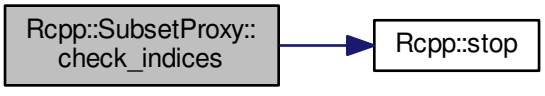
6.739.4.1 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices (int * *x*, int *n*, int *size*) [inline], [private]

Definition at line 135 of file Subsetter.h.

References Rcpp::stop().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

Here is the call graph for this function:



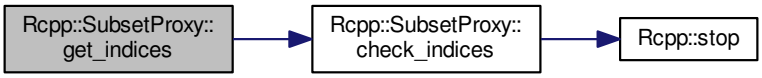
6.739.4.2 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (traits::identity< traits::int2type< INTSXP > > *t*) [inline], [private]

Definition at line 146 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, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::SubsetProxy().

Here is the call graph for this function:

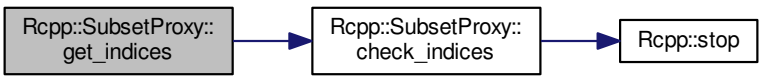


6.739.4.3 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (traits::identity< traits::int2type< REALSXP > > *t*) [inline], [private]

Definition at line 156 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, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n.

Here is the call graph for this function:

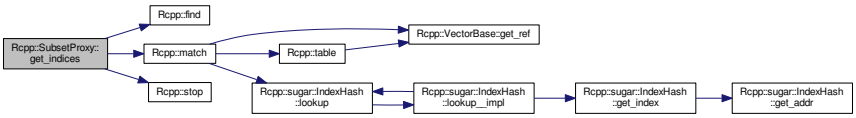


6.739.4.4 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (traits::identity< traits::int2type< STRSXP > > *t*) [inline], [private]

Definition at line 168 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:

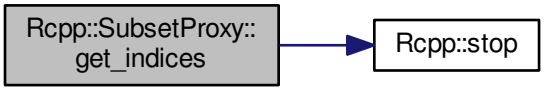


6.739.4.5 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (traits::identity< traits::int2type< LGLSXP > > *t*) [inline], [private]

Definition at line 183 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:



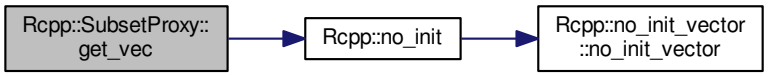
6.739.4.6 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> Vector<RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec ( ) const [inline], [private]

Definition at line 200 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:

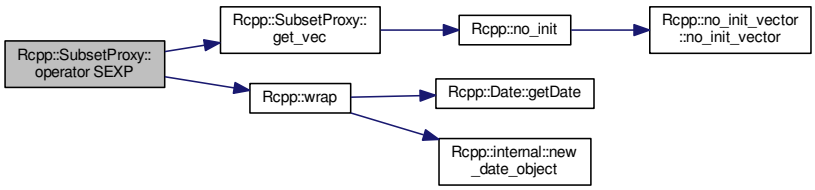


6.739.4.7 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator SEXP ( ) const [inline]

Definition at line 128 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:

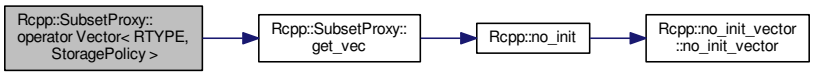


6.739.4.8 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator Vector< RTYPE, StoragePolicy > ( ) const [inline]

Definition at line 124 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:

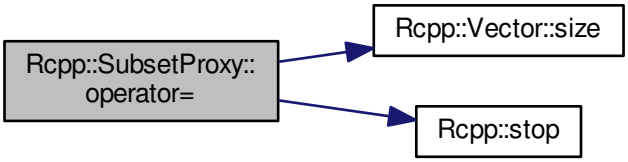


6.739.4.9 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> template<int OtherRTYPE, template< class > class OtherStoragePolicy> SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (const Vector< OtherRTYPE, OtherStoragePolicy > & *other*) [inline]

Definition at line 53 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:



6.739.4.10 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 72 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.

6.739.4.11 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 79 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.

6.739.4.12 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 86 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.

6.739.4.13 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 93 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.

6.739.4.14 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]

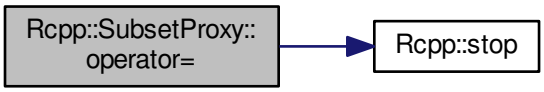
Definition at line 101 of file Subsetter.h.

6.739.4.15 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 108 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:



6.739.5 Member Data Documentation

6.739.5.1 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> std::vector<int> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices [private]

Definition at line 222 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=().

6.739.5.2 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> int Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n [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=().

6.739.5.3 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 217 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=().

6.739.5.4 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> int Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n [private]

Definition at line 219 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

6.739.5.5 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 218 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

6.739.5.6 template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T> int Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n [private]

Definition at line 220 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

6.740 tinyformat::detail::is_convertible< T1, T2 >::succeed Struct Reference

Public Attributes

- char dummy

6.740.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::succeed

Definition at line 190 of file tinyformat.h.

6.740.2 Member Data Documentation

6.740.2.1 template<typename T1 , typename T2 > char tinyformat::detail::is_convertible< T1, T2 >::succeed::dummy

Definition at line 190 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat.h

6.741 Rcpp::sugar::sugar_const_iterator_type< T > Struct Template Reference

`#include <iterator.h>`

Public Types

- typedef SugarIterator< T > type

6.741.1 Detailed Description

template<typename T>
struct Rcpp::sugar::sugar_const_iterator_type< T >

Definition at line 110 of file iterator.h.

6.741.2 Member Typedef Documentation

6.741.2.1 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

6.742 Rcpp::sugar::sugar_const_iterator_type< CharacterVector > Struct Template Reference

`#include <iterator.h>`

Public Types

- typedef SEXP * type

6.742.1 Detailed Description

template<>
struct Rcpp::sugar::sugar_const_iterator_type< CharacterVector >

Definition at line 116 of file iterator.h.

6.742.2 Member Typedef Documentation

6.742.2.1 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

6.743 Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > > Struct Template Reference

`#include <iterator.h>`

Public Types

- typedef Rcpp::Vector< RTYPE >::const_iterator type

6.743.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > >

Definition at line 113 of file iterator.h.

6.743.2 Member Typedef Documentation

6.743.2.1 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.

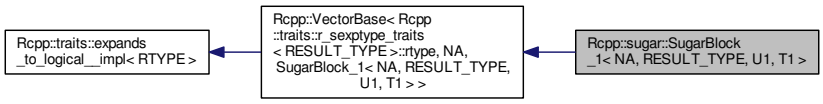
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

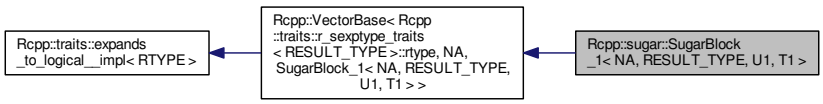
6.744 Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 > Class Template Reference

`#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

6.744.1 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.

6.744.2 Member Typedef Documentation

6.744.2.1 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.

6.744.3 Constructor & Destructor Documentation

6.744.3.1 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.

6.744.4 Member Function Documentation

6.744.4.1 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.

6.744.4.2 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.

6.744.5 Member Data Documentation

6.744.5.1 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[]().

6.744.5.2 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().

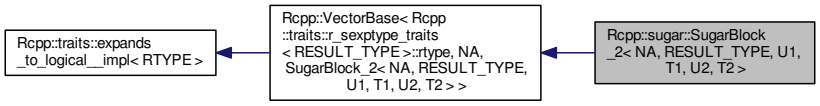
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_1.h

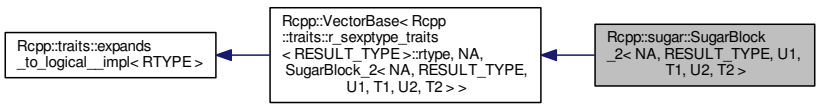
6.745 Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > Class Template Reference

`#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

6.745.1 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.

6.745.2 Member Typedef Documentation

6.745.2.1 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.

6.745.3 Constructor & Destructor Documentation

6.745.3.1 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.

6.745.4 Member Function Documentation

6.745.4.1 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.

6.745.4.2 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.

6.745.5 Member Data Documentation

6.745.5.1 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[ ](), Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ](), and Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[ ]().

6.745.5.2 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[ ](), Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ](), Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[ ](), Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::size(), and Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::size().

6.745.5.3 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[ ](), Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ](), 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().

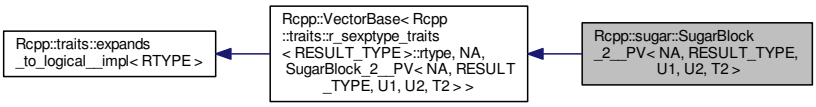
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

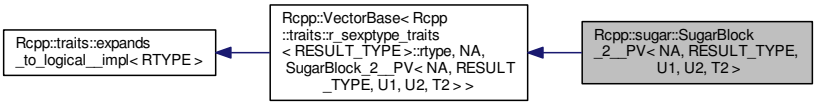
6.746 Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > Class Template Reference

`#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

6.746.1 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.

6.746.2 Member Typedef Documentation

6.746.2.1 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.

6.746.3 Constructor & Destructor Documentation

6.746.3.1 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.

6.746.4 Member Function Documentation

6.746.4.1 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< 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.

6.746.4.2 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< NA, RESULT_TYPE, U1, T1, U2, T2 >::y.

6.746.5 Member Data Documentation

6.746.5.1 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.

6.746.5.2 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.

6.746.5.3 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.

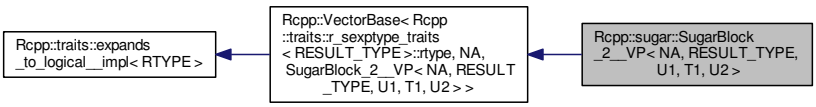
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

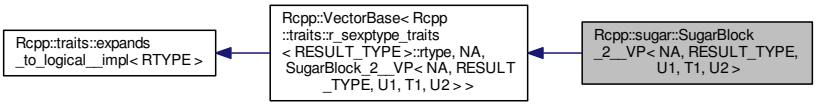
6.747 Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > Class Template Reference

`#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

6.747.1 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.

6.747.2 Member Typedef Documentation

6.747.2.1 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.

6.747.3 Constructor & Destructor Documentation

6.747.3.1 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.

6.747.4 Member Function Documentation

6.747.4.1 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< 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.

6.747.4.2 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< NA, RESULT_TYPE, U1, T1, U2, T2 >::x.

6.747.5 Member Data Documentation

6.747.5.1 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.

6.747.5.2 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.

6.747.5.3 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.

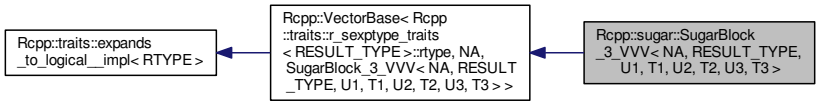
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

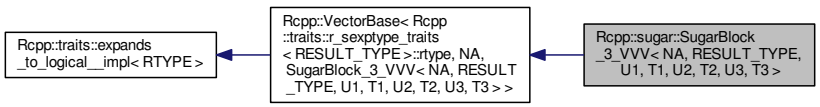
6.748 Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > Class Template Reference

`#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

6.748.1 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.

6.748.2 Member Typedef Documentation

6.748.2.1 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.

6.748.3 Constructor & Destructor Documentation

6.748.3.1 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.

6.748.4 Member Function Documentation

6.748.4.1 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.

6.748.4.2 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.

6.748.5 Member Data Documentation

6.748.5.1 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[ ]().

6.748.5.2 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().

6.748.5.3 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[ ]().

6.748.5.4 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[]().

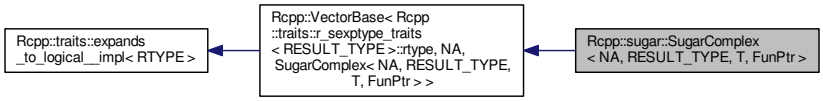
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_3.h

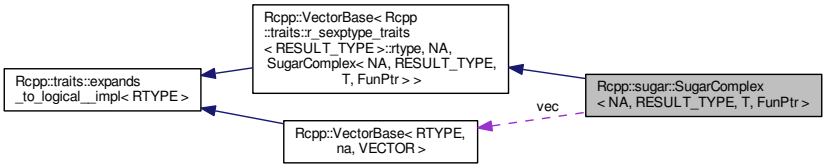
6.749 Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr > Class Template Reference

`#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

6.749.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename T, typename FunPtr>
class Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >

Definition at line 36 of file complex.h.

6.749.2 Member Typedef Documentation

6.749.2.1 template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr > typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::VEC_TYPE

Definition at line 43 of file complex.h.

6.749.3 Constructor & Destructor Documentation

6.749.3.1 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 45 of file complex.h.

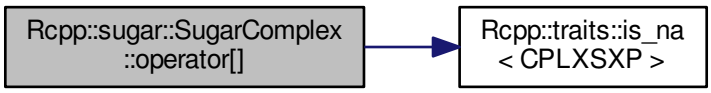
6.749.4 Member Function Documentation

6.749.4.1 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 47 of file complex.h.

References Rcpp::traits::is_na< CPLXSXP >(), 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:

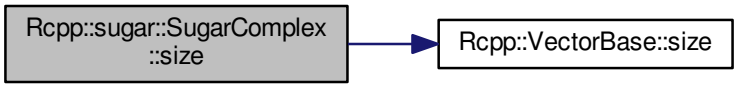


6.749.4.2 template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr > R_xlen_t Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size ( ) const [inline]

Definition at line 53 of file complex.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarComplex< NA, RESULT_T↩YPE, T, FunPtr >::vec.

Here is the call graph for this function:



6.749.5 Member Data Documentation

6.749.5.1 template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr > FunPtr Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::ptr [private]

Definition at line 56 of file complex.h.

Referenced by Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[ ]().

6.749.5.2 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 57 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

6.750 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

6.750.1 Detailed Description

template<typename T>
class Rcpp::sugar::SugarIterator< T >

Definition at line 30 of file iterator.h.

6.750.2 Member Typedef Documentation

6.750.2.1 template<typename T > typedef R_xlen_t Rcpp::sugar::SugarIterator< T >::difference_type

Definition at line 33 of file iterator.h.

6.750.2.2 template<typename T > typedef SugarIterator Rcpp::sugar::SugarIterator< T >::iterator

Definition at line 38 of file iterator.h.

6.750.2.3 template<typename T > typedef std::random_access_iterator_tag Rcpp::sugar::SugarIterator< T >::iterator_category

Definition at line 37 of file iterator.h.

6.750.2.4 template<typename T > typedef STORAGE_TYPE* Rcpp::sugar::SugarIterator< T >::pointer

Definition at line 36 of file iterator.h.

6.750.2.5 template<typename T > typedef STORAGE_TYPE Rcpp::sugar::SugarIterator< T >::reference

Definition at line 35 of file iterator.h.

6.750.2.6 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.

6.750.3 Constructor & Destructor Documentation

6.750.3.1 template<typename T > Rcpp::sugar::SugarIterator< T >::SugarIterator (const T & *ref_*) [inline]

Definition at line 40 of file iterator.h.

6.750.3.2 template<typename T > Rcpp::sugar::SugarIterator< T >::SugarIterator (const T & *ref_*, R_xlen_t *index_*) [inline]

Definition at line 41 of file iterator.h.

6.750.3.3 template<typename T > Rcpp::sugar::SugarIterator< T >::SugarIterator (const SugarIterator< T > & *other*) [inline]

Definition at line 42 of file iterator.h.

6.750.4 Member Function Documentation

6.750.4.1 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.

6.750.4.2 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.

6.750.4.3 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.

6.750.4.4 template<typename T > iterator& Rcpp::sugar::SugarIterator< T >::operator++ ( ) [inline]

Definition at line 44 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.750.4.5 template<typename T > iterator Rcpp::sugar::SugarIterator< T >::operator++ (int) [inline]

Definition at line 45 of file iterator.h.

6.750.4.6 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.

6.750.4.7 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.

6.750.4.8 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.

6.750.4.9 template<typename T > iterator& Rcpp::sugar::SugarIterator< T >::operator-- ( ) [inline]

Definition at line 50 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.750.4.10 template<typename T > iterator Rcpp::sugar::SugarIterator< T >::operator-- (int) [inline]

Definition at line 51 of file iterator.h.

6.750.4.11 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.

6.750.4.12 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.

6.750.4.13 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.

6.750.4.14 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.

6.750.4.15 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.

6.750.4.16 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.

6.750.4.17 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.

6.750.4.18 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.

6.750.5 Member Data Documentation

6.750.5.1 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[ ]().

6.750.5.2 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[ ]().

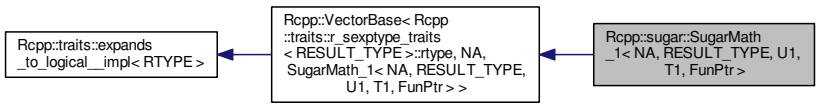
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

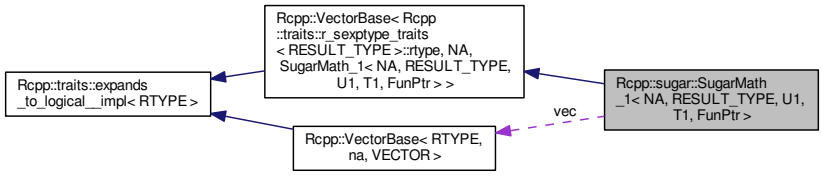
6.751 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > Class Template Reference

`#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

6.751.1 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.

6.751.2 Member Typedef Documentation

6.751.2.1 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.

6.751.3 Constructor & Destructor Documentation

6.751.3.1 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.

6.751.4 Member Function Documentation

6.751.4.1 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.

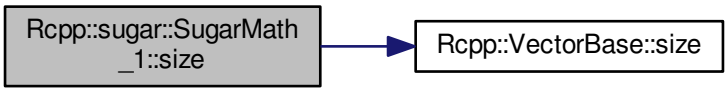
References Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

6.751.4.2 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:



6.751.5 Member Data Documentation

6.751.5.1 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[ ]().

6.751.5.2 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().

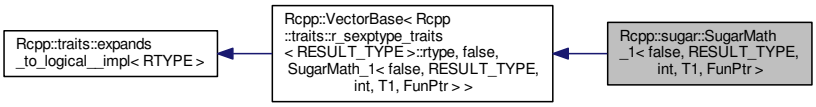
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

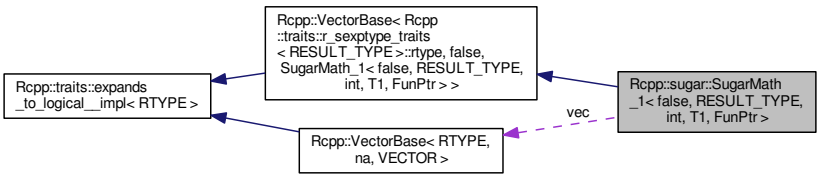
6.752 Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > Class Template Reference

`#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

6.752.1 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.

6.752.2 Member Typedef Documentation

6.752.2.1 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.

6.752.3 Constructor & Destructor Documentation

6.752.3.1 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.

6.752.4 Member Function Documentation

6.752.4.1 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.

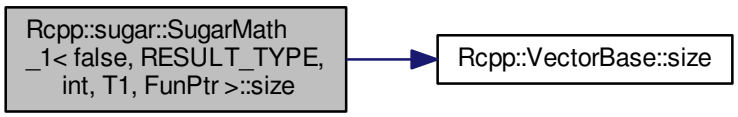
References Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

6.752.4.2 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:



6.752.5 Member Data Documentation

6.752.5.1 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.

6.752.5.2 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.

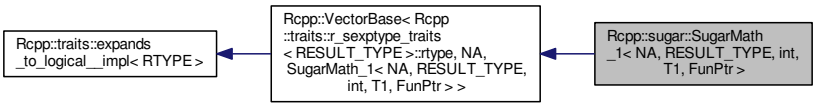
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

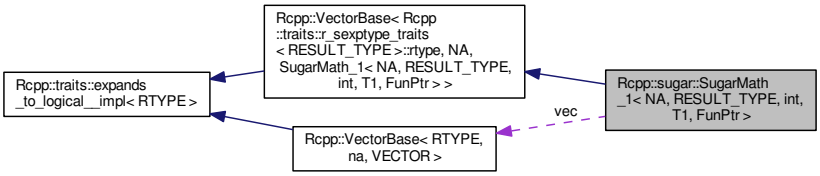
6.753 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > Class Template Reference

`#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

6.753.1 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.

6.753.2 Member Typedef Documentation

6.753.2.1 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.

6.753.3 Constructor & Destructor Documentation

6.753.3.1 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.

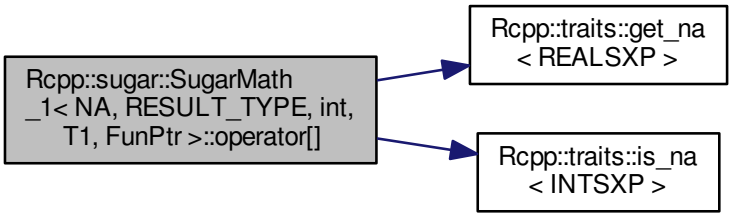
6.753.4 Member Function Documentation

6.753.4.1 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:

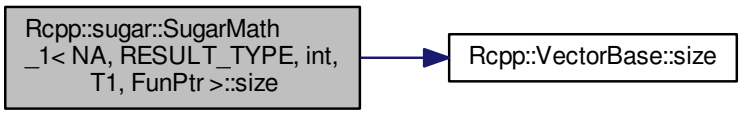


6.753.4.2 template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr > R_xlen_t Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size ( ) const [inline]

Definition at line 68 of file SugarMath.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarMath_1< NA, RESULT_T↩YPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:



6.753.5 Member Data Documentation

6.753.5.1 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.

6.753.5.2 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.

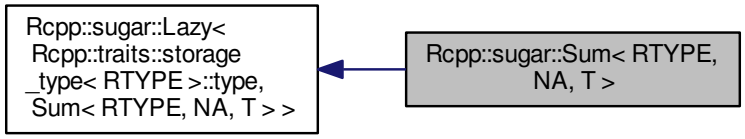
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

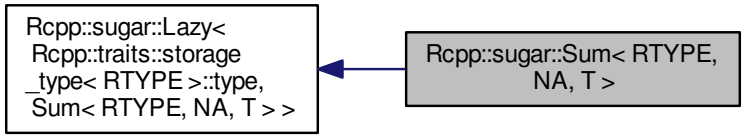
6.754 Rcpp::sugar::Sum< RTYPE, NA, T > Class Template Reference

`#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

6.754.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sum< RTYPE, NA, T >

Definition at line 29 of file sum.h.

6.754.2 Member Typedef Documentation

6.754.2.1 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.

6.754.2.2 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.

6.754.2.3 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.

6.754.3 Constructor & Destructor Documentation

6.754.3.1 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.

6.754.4 Member Function Documentation

6.754.4.1 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.

6.754.5 Member Data Documentation

6.754.5.1 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.

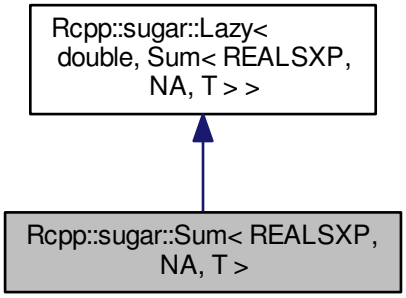
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

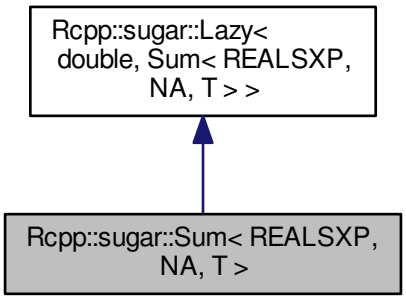
6.755 Rcpp::sugar::Sum< REALSXP, NA, T > Class Template Reference

`#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

6.755.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Sum< REALSXP, NA, T >

Definition at line 54 of file sum.h.

6.755.2 Member Typedef Documentation

6.755.2.1 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.

6.755.2.2 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.

6.755.3 Constructor & Destructor Documentation

6.755.3.1 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.

6.755.4 Member Function Documentation

6.755.4.1 template<bool NA, typename T > double Rcpp::sugar::Sum< REALSXP, NA, T >::get ( ) const [inline]

Definition at line 61 of file sum.h.

6.755.5 Member Data Documentation

6.755.5.1 template<bool NA, typename T > const VEC_EXT& Rcpp::sugar::Sum< REALSXP, NA, T >::object [private]

Definition at line 70 of file sum.h.

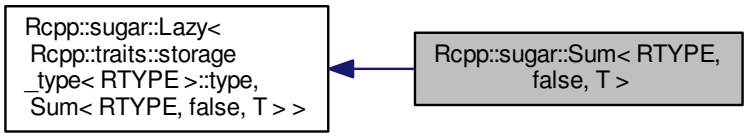
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

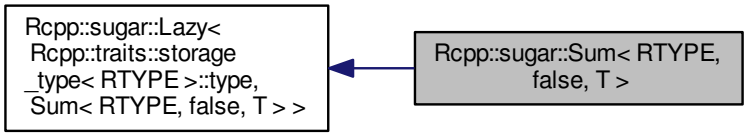
6.756 Rcpp::sugar::Sum< RTYPE, false, T > Class Template Reference

`#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

6.756.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Sum< RTYPE, false, T >

Definition at line 75 of file sum.h.

6.756.2 Member Typedef Documentation

6.756.2.1 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.

6.756.2.2 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.

6.756.2.3 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.

6.756.3 Constructor & Destructor Documentation

6.756.3.1 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.

6.756.4 Member Function Documentation

6.756.4.1 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Sum< RTYPE, false, T >::get ( ) const [inline]

Definition at line 83 of file sum.h.

6.756.5 Member Data Documentation

6.756.5.1 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

6.757 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

6.757.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::Table< RTYPE, TABLE_T >

Definition at line 89 of file table.h.

6.757.2 Member Typedef Documentation

6.757.2.1 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.

6.757.2.2 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.

6.757.2.3 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.

6.757.2.4 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.

6.757.3 Constructor & Destructor Documentation

6.757.3.1 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::sugar::CountInserter< HASH, STORAGE >::hash.

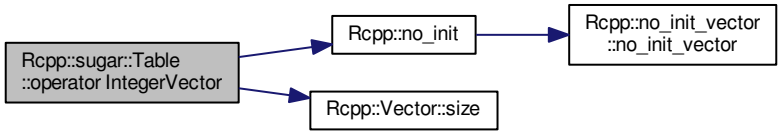
6.757.4 Member Function Documentation

6.757.4.1 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::Grabber< HASH, RTYPE >::names, Rcpp::no_init(), and Rcpp::Vector< RTYPE, Storage↩ Policy >::size().

Here is the call graph for this function:



6.757.5 Member Data Documentation

6.757.5.1 template<int RTYPE, typename TABLE_T > HASH Rcpp::sugar::Table< RTYPE, TABLE_T >::hash [private]

Definition at line 114 of file table.h.

6.757.5.2 template<int RTYPE, typename TABLE_T > SORTED_MAP Rcpp::sugar::Table< RTYPE, TABLE_T >::map [private]

Definition at line 117 of file table.h.

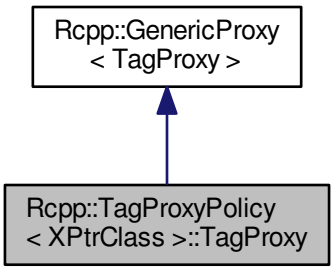
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

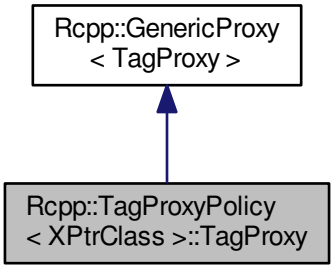
6.758 Rcpp::TagProxyPolicy< XPtrClass >::TagProxy Class Reference

`#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

6.758.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::TagProxy

Definition at line 27 of file TagProxy.h.

6.758.2 Constructor & Destructor Documentation

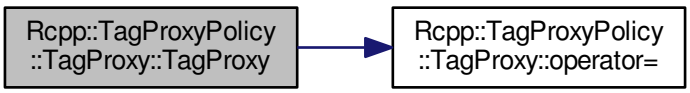
6.758.2.1 template<typename XPtrClass> Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::TagProxy (XPtrClass & *xp_*) [inline]

Definition at line 29 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator=().

Referenced by Rcpp::TagProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::tag().

Here is the call graph for this function:



6.758.3 Member Function Documentation

6.758.3.1 template<typename XPtrClass> SEXP Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::get ( ) const [inline], [private]

Definition at line 43 of file TagProxy.h.

6.758.3.2 template<typename CLASS > Rcpp::TagProxyPolicy< CLASS >::TagProxy::operator SEXP ( ) const

Definition at line 111 of file proxy.h.

6.758.3.3 template<typename XPtrClass> template<typename U > Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator U ( ) const

6.758.3.4 template<typename XPtrClass> template<typename U > TagProxy& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator= (const U & *u*)

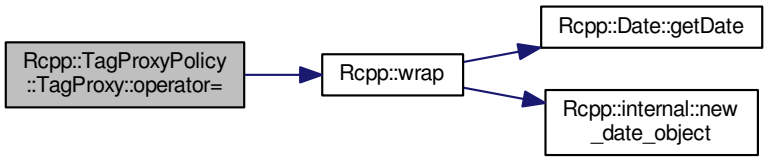
Referenced by Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::TagProxy().

6.758.3.5 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:



6.758.3.6 template<typename XPtrClass> void Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::set (SEXP *x*) [inline], [private]

Definition at line 47 of file TagProxy.h.

6.758.4 Member Data Documentation

6.758.4.1 template<typename XPtrClass> XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp [private]

Definition at line 41 of file TagProxy.h.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/TagProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.759 Rcpp::TagProxyPolicy< XPtrClass > Class Template Reference

`#include <TagProxy.h>`

Classes

- class const_TagProxy
- class TagProxy

Public Member Functions

- TagProxy tag ()
- const_TagProxy tag () const

6.759.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >

Definition at line 24 of file TagProxy.h.

6.759.2 Member Function Documentation

6.759.2.1 template<typename XPtrClass> TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( ) [inline]

Definition at line 73 of file TagProxy.h.

6.759.2.2 template<typename XPtrClass> const_TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( ) const [inline]

Definition at line 77 of file TagProxy.h.

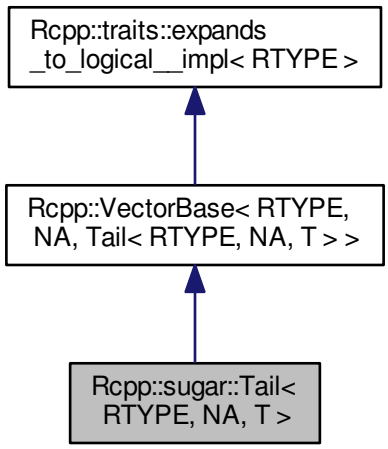
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/TagProxy.h

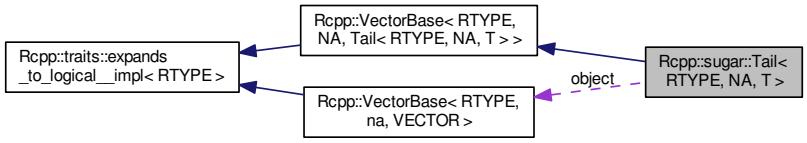
6.760 Rcpp::sugar::Tail< RTYPE, NA, T > Class Template Reference

`#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

6.760.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Tail< RTYPE, NA, T >

Definition at line 29 of file tail.h.

6.760.2 Member Typedef Documentation

6.760.2.1 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.

6.760.2.2 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.

6.760.3 Constructor & Destructor Documentation

6.760.3.1 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.

6.760.4 Member Function Documentation

6.760.4.1 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.

6.760.4.2 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.

6.760.5 Member Data Documentation

6.760.5.1 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(), Rcpp::sugar::Tail< RTYPE, NA, T >::Tail(), and Rcpp::tail().

6.760.5.2 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.

6.760.5.3 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().

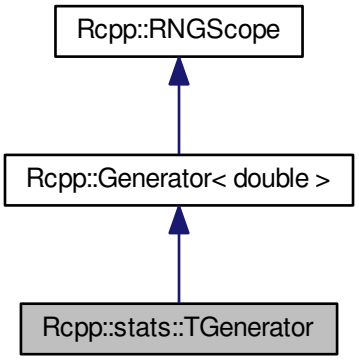
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/tail.h

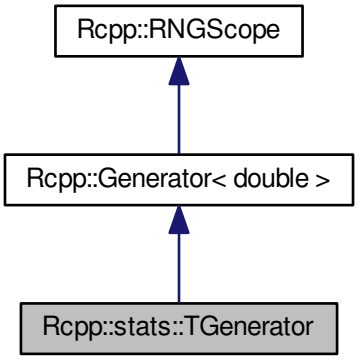
6.761 Rcpp::stats::TGenerator Class Reference

```
#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

6.761.1 Detailed Description

Definition at line 28 of file rt.h.

6.761.2 Constructor & Destructor Documentation

6.761.2.1 Rcpp::stats::TGenerator::TGenerator (double *df_*) [inline]

Definition at line 31 of file rt.h.

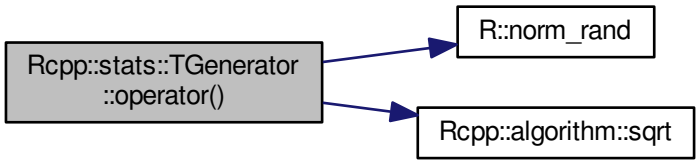
6.761.3 Member Function Documentation

6.761.3.1 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:



6.761.4 Member Data Documentation

6.761.4.1 double Rcpp::stats::TGenerator::df [private]

Definition at line 45 of file rt.h.

Referenced by operator()().

6.761.4.2 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

6.762 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

6.762.1 Detailed Description

Examples:

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 31 of file Timer.h.

6.762.2 Constructor & Destructor Documentation

6.762.2.1 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:

6.762.3 Member Function Documentation

6.762.3.1 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().

6.762.3.2 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().

6.762.3.3 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().

6.762.3.4 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().

6.762.3.5 void Timer::Start ( ) [inline]

Examples:

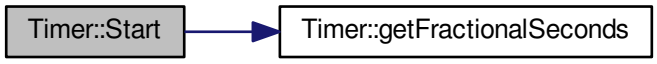
SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 34 of file Timer.h.

References getFractionalSeconds(), and start_t.

Referenced by main().

Here is the call graph for this function:



6.762.3.6 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:

6.762.4 Member Data Documentation

6.762.4.1 double Timer::cumul [private]

Examples:

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by CumulativeTime(), Reset(), and Stop().

6.762.4.2 double Timer::elapsed [private]

Examples:

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by ElapsedTime(), Reset(), and Stop().

6.762.4.3 double Timer::end_t [private]

Examples:

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by Reset(), and Stop().

6.762.4.4 double Timer::start_t [private]

Examples:

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by Reset(), Start(), and Stop().

6.762.4.5 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

6.763 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

6.763.1 Detailed Description

Definition at line 103 of file Timer.h.

6.763.2 Member Typedef Documentation

6.763.2.1 typedef std::pair<std::string,nanotime_t> Rcpp::Timer::Step [private]

Definition at line 137 of file Timer.h.

6.763.2.2 typedef std::vector<Step> Rcpp::Timer::Steps [private]

Definition at line 138 of file Timer.h.

6.763.3 Constructor & Destructor Documentation

6.763.3.1 Rcpp::Timer::Timer ( ) [inline]

Definition at line 105 of file Timer.h.

Referenced by get_timers().

6.763.3.2 Rcpp::Timer::Timer (nanotime_t *start_time_*) [inline]

Definition at line 106 of file Timer.h.

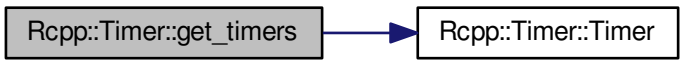
6.763.4 Member Function Documentation

6.763.4.1 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:



6.763.4.2 nanotime_t Rcpp::Timer::now ( ) const [inline]

Definition at line 128 of file Timer.h.

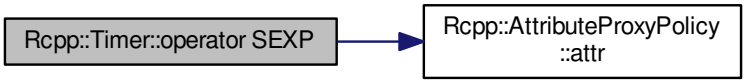
Referenced by step().

6.763.4.3 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:



6.763.4.4 nanotime_t Rcpp::Timer::origin ( ) const [inline]

Definition at line 132 of file Timer.h.

References start_time.

6.763.4.5 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:

6.763.5 Member Data Documentation

6.763.5.1 Steps Rcpp::Timer::data [private]

Definition at line 140 of file Timer.h.

Referenced by operator SEXP(), and step().

6.763.5.2 const nanotime_t Rcpp::Timer::start_time [private]

Definition at line 141 of file Timer.h.

Referenced by operator SEXP(), and origin().

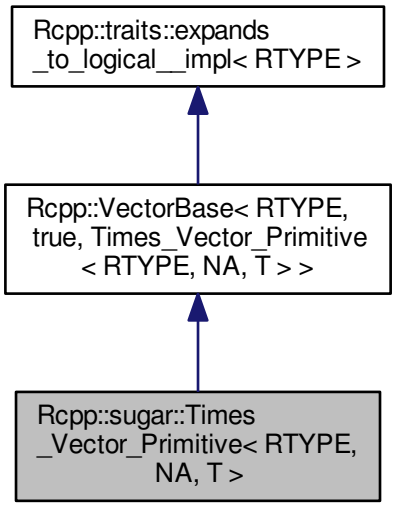
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Benchmark/Timer.h

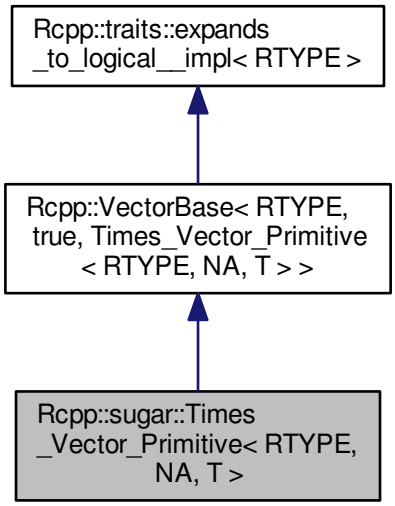
6.764 Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T > Class Template Reference

`#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

6.764.1 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.

6.764.2 Member Typedef Documentation

6.764.2.1 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.

6.764.2.2 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.

6.764.2.3 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.

6.764.3 Constructor & Destructor Documentation

6.764.3.1 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.

6.764.4 Member Function Documentation

6.764.4.1 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_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.

6.764.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.764.5 Member Data Documentation

6.764.5.1 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.

6.764.5.2 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.

6.764.5.3 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.

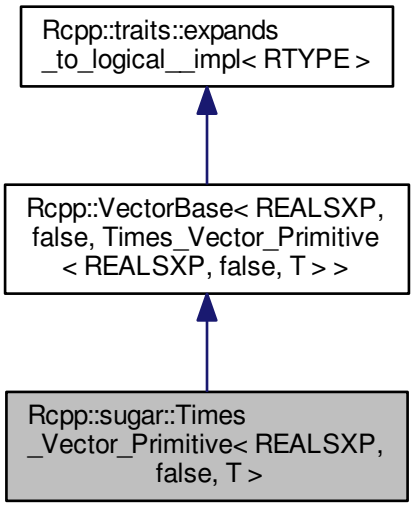
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

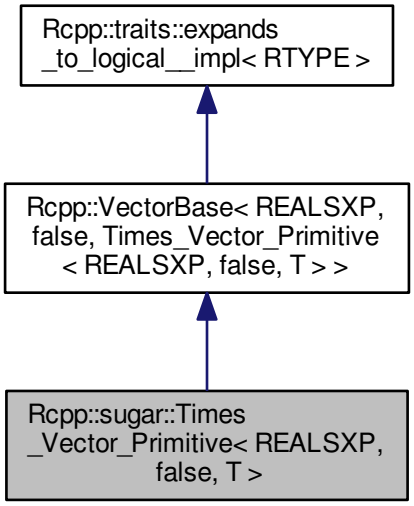
6.765 Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T > Class Template Reference

`#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

6.765.1 Detailed Description

template<typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >

Definition at line 309 of file times.h.

6.765.2 Member Typedef Documentation

6.765.2.1 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.

6.765.2.2 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.

6.765.3 Constructor & Destructor Documentation

6.765.3.1 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.

6.765.4 Member Function Documentation

6.765.4.1 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_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.

6.765.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.765.5 Member Data Documentation

6.765.5.1 template<typename T > const EXT& Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::lhs [private]

Definition at line 325 of file times.h.

6.765.5.2 template<typename T > double Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::rhs [private]

Definition at line 326 of file times.h.

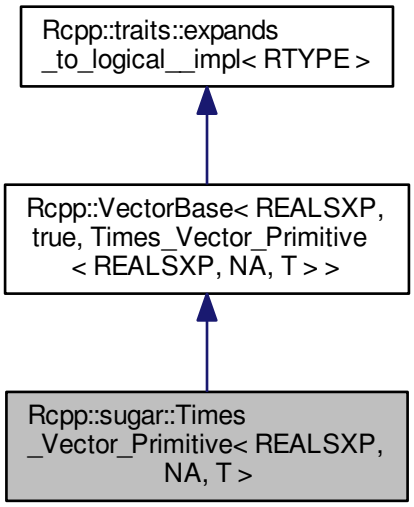
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

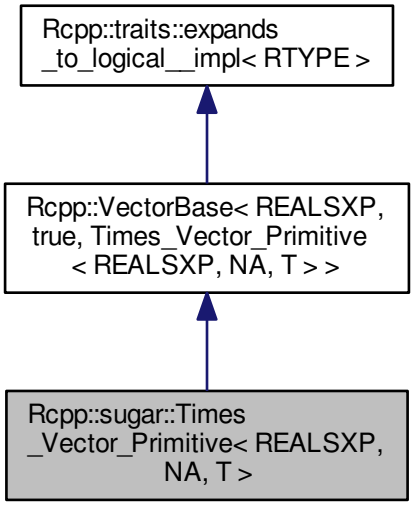
6.766 Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T > Class Template Reference

`#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

6.766.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >

Definition at line 262 of file times.h.

6.766.2 Member Typedef Documentation

6.766.2.1 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.

6.766.2.2 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.

6.766.3 Constructor & Destructor Documentation

6.766.3.1 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.

6.766.4 Member Function Documentation

6.766.4.1 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_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.

6.766.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.766.5 Member Data Documentation

6.766.5.1 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.

6.766.5.2 template<bool NA, typename T > double Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::rhs [private]

Definition at line 280 of file times.h.

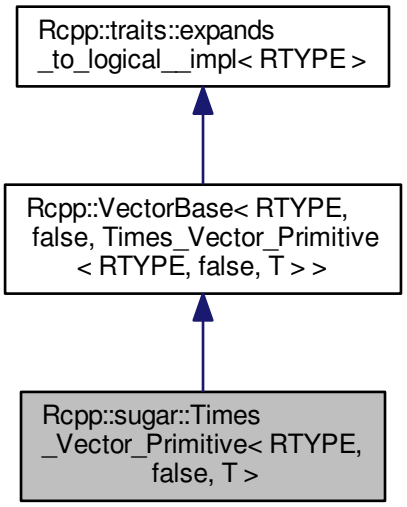
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

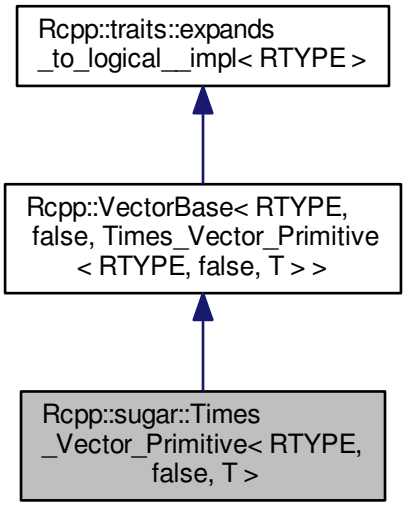
6.767 Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T > Class Template Reference

`#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

6.767.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >

Definition at line 285 of file times.h.

6.767.2 Member Typedef Documentation

6.767.2.1 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.

6.767.2.2 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.

6.767.2.3 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.

6.767.3 Constructor & Destructor Documentation

6.767.3.1 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.

6.767.4 Member Function Documentation

6.767.4.1 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_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.

6.767.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.767.5 Member Data Documentation

6.767.5.1 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.

6.767.5.2 template<int RTYPE, typename T > STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::rhs [private]

Definition at line 303 of file times.h.

6.767.5.3 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.

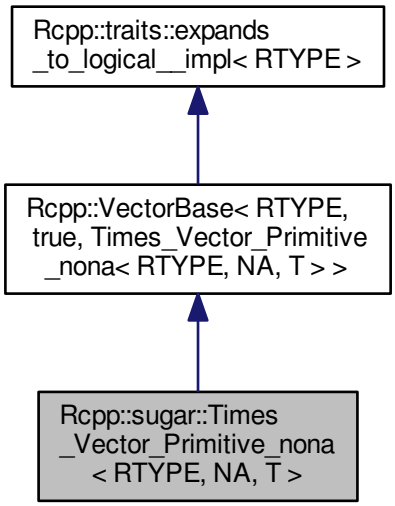
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

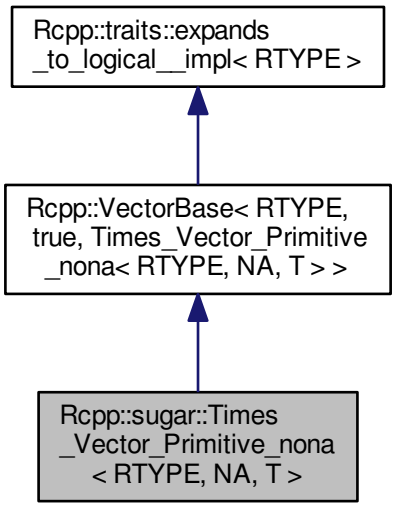
6.768 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference

`#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

6.768.1 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.

6.768.2 Member Typedef Documentation

6.768.2.1 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.

6.768.2.2 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.

6.768.2.3 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.

6.768.3 Constructor & Destructor Documentation

6.768.3.1 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.

6.768.4 Member Function Documentation

6.768.4.1 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_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.

6.768.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.768.5 Member Data Documentation

6.768.5.1 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.

6.768.5.2 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.

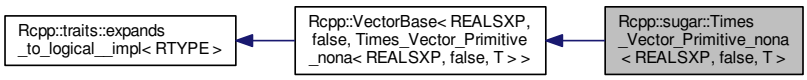
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

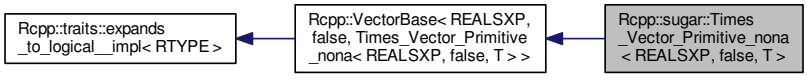
6.769 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T > Class Template Reference

`#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 >::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

6.769.1 Detailed Description

template<typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >

Definition at line 405 of file times.h.

6.769.2 Member Typedef Documentation

6.769.2.1 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.

6.769.2.2 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.

6.769.3 Constructor & Destructor Documentation

6.769.3.1 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.

6.769.4 Member Function Documentation

6.769.4.1 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_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.

6.769.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.769.5 Member Data Documentation

6.769.5.1 template<typename T > const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::lhs [private]

Definition at line 421 of file times.h.

6.769.5.2 template<typename T > double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::rhs [private]

Definition at line 422 of file times.h.

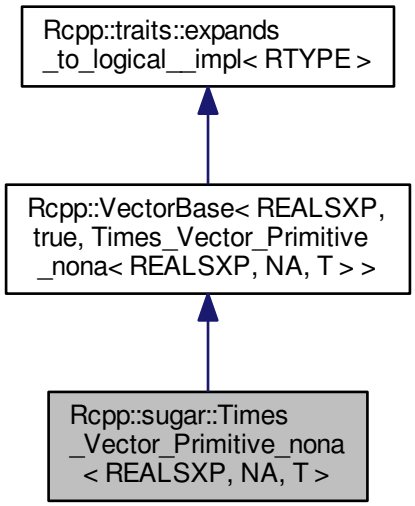
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

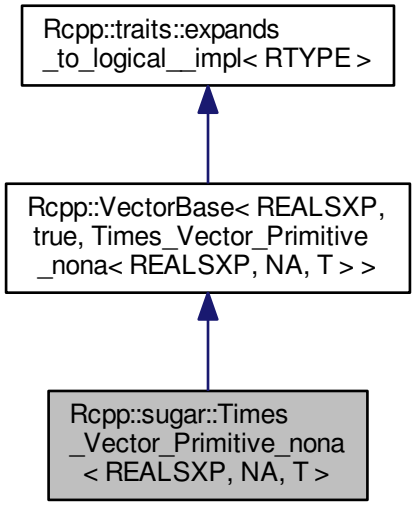
6.770 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T > Class Template Reference

`#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

6.770.1 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.

6.770.2 Member Typedef Documentation

6.770.2.1 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.

6.770.2.2 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.

6.770.3 Constructor & Destructor Documentation

6.770.3.1 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.

6.770.4 Member Function Documentation

6.770.4.1 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_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.

6.770.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.770.5 Member Data Documentation

6.770.5.1 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.

6.770.5.2 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.

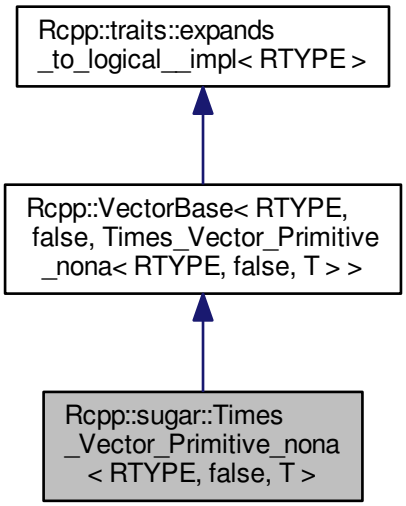
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

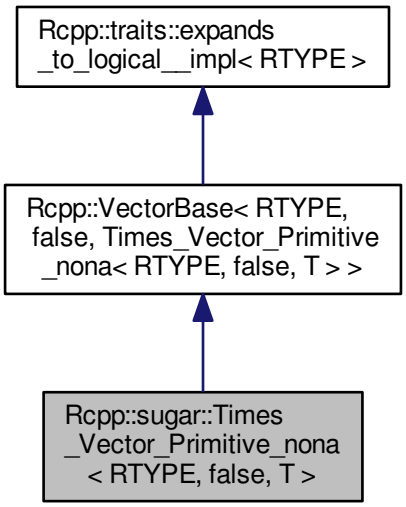
6.771 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference

`#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

6.771.1 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.

6.771.2 Member Typedef Documentation

6.771.2.1 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.

6.771.2.2 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.

6.771.2.3 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.

6.771.3 Constructor & Destructor Documentation

6.771.3.1 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.

6.771.4 Member Function Documentation

6.771.4.1 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_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.

6.771.4.2 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_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.771.5 Member Data Documentation

6.771.5.1 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.

6.771.5.2 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.

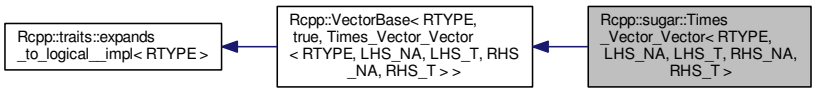
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

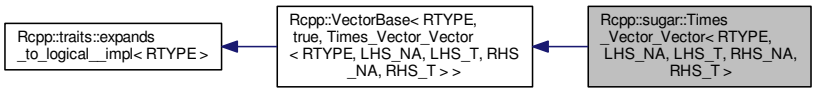
6.772 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.772.1 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.

6.772.2 Member Typedef Documentation

6.772.2.1 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.

6.772.2.2 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.

6.772.2.3 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.

6.772.2.4 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.

6.772.2.5 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.

6.772.3 Constructor & Destructor Documentation

6.772.3.1 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.

6.772.4 Member Function Documentation

6.772.4.1 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.

6.772.4.2 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.

6.772.5 Member Data Documentation

6.772.5.1 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::operator∗(), 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_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_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_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< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size(), 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(), Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::size(), 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().

6.772.5.2 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::operator*(), 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_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_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[]().

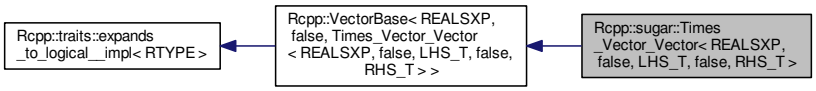
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

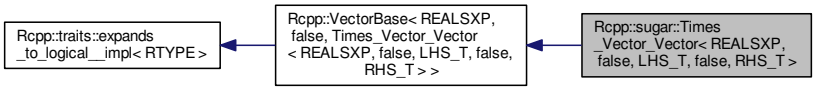
6.773 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.773.1 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.

6.773.2 Member Typedef Documentation

6.773.2.1 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.

6.773.2.2 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.

6.773.2.3 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.

6.773.2.4 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.

6.773.3 Constructor & Destructor Documentation

6.773.3.1 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.

6.773.4 Member Function Documentation

6.773.4.1 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.

6.773.4.2 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.

6.773.5 Member Data Documentation

6.773.5.1 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.

6.773.5.2 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.

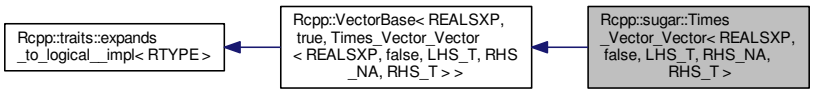
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

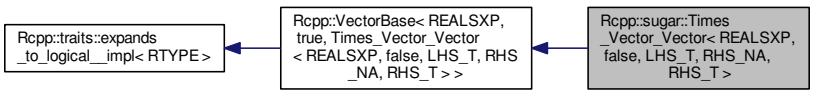
6.774 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.774.1 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.

6.774.2 Member Typedef Documentation

6.774.2.1 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.

6.774.2.2 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.

6.774.2.3 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.

6.774.2.4 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.

6.774.3 Constructor & Destructor Documentation

6.774.3.1 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.

6.774.4 Member Function Documentation

6.774.4.1 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.

6.774.4.2 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.

6.774.5 Member Data Documentation

6.774.5.1 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.

6.774.5.2 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.

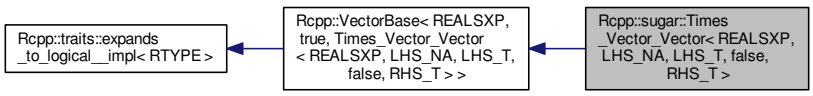
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

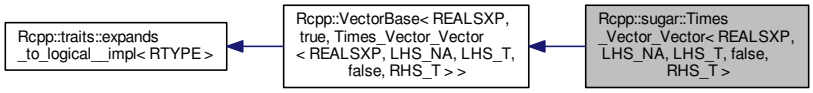
6.775 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.775.1 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.

6.775.2 Member Typedef Documentation

6.775.2.1 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.

6.775.2.2 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.

6.775.2.3 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.

6.775.2.4 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.

6.775.3 Constructor & Destructor Documentation

6.775.3.1 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.

6.775.4 Member Function Documentation

6.775.4.1 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.

6.775.4.2 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.

6.775.5 Member Data Documentation

6.775.5.1 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.

6.775.5.2 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.

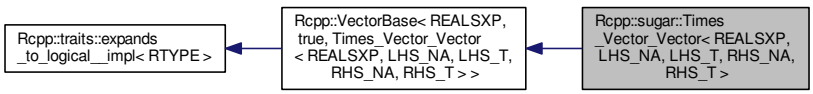
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

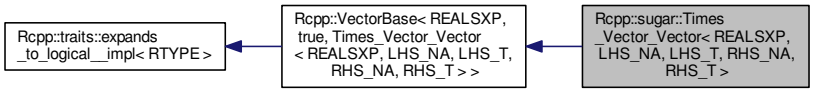
6.776 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.776.1 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.

6.776.2 Member Typedef Documentation

6.776.2.1 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.

6.776.2.2 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.

6.776.2.3 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.

6.776.2.4 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.

6.776.3 Constructor & Destructor Documentation

6.776.3.1 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.

6.776.4 Member Function Documentation

6.776.4.1 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.

6.776.4.2 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.

6.776.5 Member Data Documentation

6.776.5.1 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.

6.776.5.2 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.

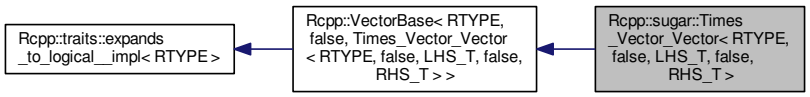
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

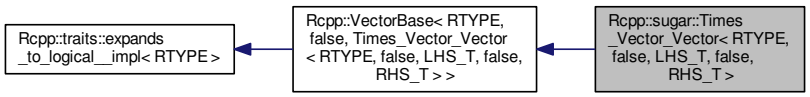
6.777 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

`#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

6.777.1 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.

6.777.2 Member Typedef Documentation

6.777.2.1 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.

6.777.2.2 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.

6.777.2.3 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.

6.777.2.4 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.

6.777.2.5 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.

6.777.3 Constructor & Destructor Documentation

6.777.3.1 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.

6.777.4 Member Function Documentation

6.777.4.1 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.

6.777.4.2 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.

6.777.5 Member Data Documentation

6.777.5.1 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.

6.777.5.2 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.

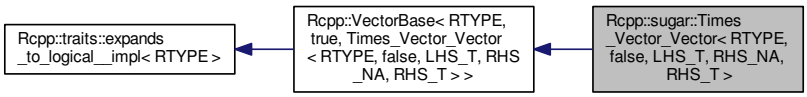
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

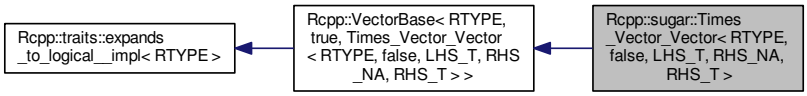
6.778 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

`#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

6.778.1 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.

6.778.2 Member Typedef Documentation

6.778.2.1 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.

6.778.2.2 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.778.2.3 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.

6.778.2.4 template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T > typedef Rcpp::VectorBase<R↩ TYPE,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.

6.778.2.5 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.

6.778.3 Constructor & Destructor Documentation

6.778.3.1 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.

6.778.4 Member Function Documentation

6.778.4.1 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.

6.778.4.2 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.

6.778.5 Member Data Documentation

6.778.5.1 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.

6.778.5.2 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.

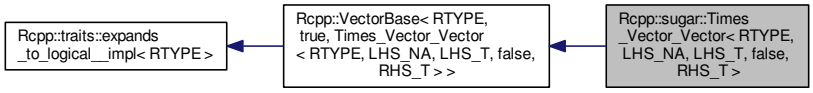
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

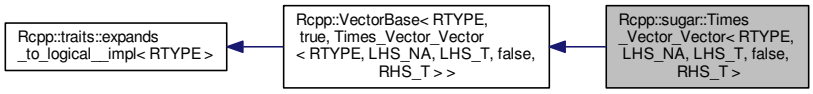
6.779 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

`#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

6.779.1 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.

6.779.2 Member Typedef Documentation

6.779.2.1 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.

6.779.2.2 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.

6.779.2.3 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.

6.779.2.4 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.

6.779.2.5 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.

6.779.3 Constructor & Destructor Documentation

6.779.3.1 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.

6.779.4 Member Function Documentation

6.779.4.1 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.

6.779.4.2 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.

6.779.5 Member Data Documentation

6.779.5.1 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.

6.779.5.2 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

6.780 Rcpp::ttinfo Struct Reference

Public Attributes

- long tt_gmtoff
- int tt_isdst
- int tt_abbrind
- int tt_ttisstd
- int tt_ttisgmt

6.780.1 Detailed Description

Definition at line 321 of file Date.cpp.

6.780.2 Member Data Documentation

6.780.2.1 int Rcpp::ttinfo::tt_abbrind

Definition at line 324 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.780.2.2 long Rcpp::ttinfo::tt_gmtoff

Definition at line 322 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.780.2.3 int Rcpp::ttinfo::tt_isdst

Definition at line 323 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.780.2.4 int Rcpp::ttinfo::tt_ttisgmt

Definition at line 326 of file Date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.780.2.5 int Rcpp::ttinfo::tt_ttisstd

Definition at line 325 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

6.781 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_

6.781.1 Detailed Description

Definition at line 164 of file attributes.cpp.

6.781.2 Constructor & Destructor Documentation

6.781.2.1 Rcpp::attributes::Type::Type ( ) [inline]

Definition at line 166 of file attributes.cpp.

6.781.2.2 Rcpp::attributes::Type::Type (const std::string & *name,* bool *isConst,* bool *isReference*) [inline]

Definition at line 167 of file attributes.cpp.

6.781.3 Member Function Documentation

6.781.3.1 bool Rcpp::attributes::Type::empty ( ) const [inline]

Definition at line 171 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<(), and Rcpp::attributes::SourceFileAttributesParser::parseFunction().

6.781.3.2 std::string Rcpp::attributes::Type::full_name ( ) const [inline]

Definition at line 184 of file attributes.cpp.

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

6.781.3.3 bool Rcpp::attributes::Type::isConst ( ) const [inline]

Definition at line 193 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.781.3.4 bool Rcpp::attributes::Type::isReference ( ) const [inline]

Definition at line 194 of file attributes.cpp.

Referenced by Rcpp::attributes::operator<<().

6.781.3.5 bool Rcpp::attributes::Type::isVoid ( ) const [inline]

Definition at line 192 of file attributes.cpp.

6.781.3.6 const std::string& Rcpp::attributes::Type::name ( ) const [inline]

Definition at line 183 of file attributes.cpp.

Referenced by Rcpp::attributes::generateRArgList(), and Rcpp::attributes::operator<<().

6.781.3.7 bool Rcpp::attributes::Type::operator!= (const Type & *other*) const [inline]

Definition at line 179 of file attributes.cpp.

6.781.3.8 bool Rcpp::attributes::Type::operator== (const Type & *other*) const [inline]

Definition at line 173 of file attributes.cpp.

References isConst_, isReference_, and name_.

6.781.4 Member Data Documentation

6.781.4.1 bool Rcpp::attributes::Type::isConst_ [private]

Definition at line 198 of file attributes.cpp.

Referenced by operator==().

6.781.4.2 bool Rcpp::attributes::Type::isReference_ [private]

Definition at line 199 of file attributes.cpp.

Referenced by operator==().

6.781.4.3 std::string Rcpp::attributes::Type::name_ [private]

Definition at line 197 of file attributes.cpp.

Referenced by operator==().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.782 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]

6.782.1 Detailed Description

Definition at line 160 of file Date.cpp.

6.782.2 Member Data Documentation

6.782.2.1 char Rcpp::tzhead::tzh_charcnt[4]

Definition at line 169 of file Date.cpp.

6.782.2.2 char Rcpp::tzhead::tzh_leapcnt[4]

Definition at line 166 of file Date.cpp.

6.782.2.3 char Rcpp::tzhead::tzh_magic[4]

Definition at line 161 of file Date.cpp.

6.782.2.4 char Rcpp::tzhead::tzh_reserved[15]

Definition at line 163 of file Date.cpp.

6.782.2.5 char Rcpp::tzhead::tzh_timecnt[4]

Definition at line 167 of file Date.cpp.

6.782.2.6 char Rcpp::tzhead::tzh_ttisgmtcnt[4]

Definition at line 164 of file Date.cpp.

6.782.2.7 char Rcpp::tzhead::tzh_ttisstdcnt[4]

Definition at line 165 of file Date.cpp.

6.782.2.8 char Rcpp::tzhead::tzh_typecnt[4]

Definition at line 168 of file Date.cpp.

6.782.2.9 char Rcpp::tzhead::tzh_version[1]

Definition at line 162 of file Date.cpp.

The documentation for this struct was generated from the following file:

- src/Date.cpp

6.783 Rcpp::traits::un_pointer< T > Struct Template Reference

`#include <un_pointer.h>`

Public Types

- typedef T type

6.783.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< T >

Definition at line 29 of file un_pointer.h.

6.783.2 Member Typedef Documentation

6.783.2.1 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

6.784 Rcpp::traits::un_pointer< object< T > > Struct Template Reference

`#include <un_pointer.h>`

Public Types

- typedef T type

6.784.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< object< T > >

Definition at line 31 of file un_pointer.h.

6.784.2 Member Typedef Documentation

6.784.2.1 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

6.785 Rcpp::traits::un_pointer< T * > Struct Template Reference

`#include <un_pointer.h>`

Public Types

- typedef T type

6.785.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< T * >

Definition at line 30 of file un_pointer.h.

6.785.2 Member Typedef Documentation

6.785.2.1 template<typename T > typedef T Rcpp::traits::un_pointer< T * >::type

Definition at line 30 of file un_pointer.h.

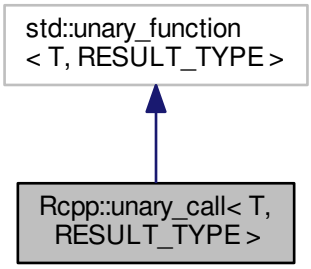
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un_pointer.h

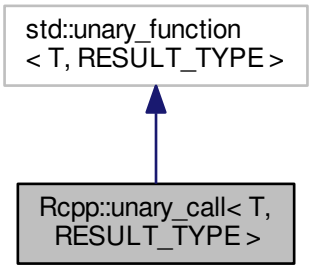
6.786 Rcpp::unary_call< T, RESULT_TYPE > Class Template Reference

`#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

6.786.1 Detailed Description

template<typename T, typename RESULT_TYPE = SEXP>
class Rcpp::unary_call< T, RESULT_TYPE >

Definition at line 180 of file Language.h.

6.786.2 Constructor & Destructor Documentation

6.786.2.1 template<typename T , typename RESULT_TYPE = SEXP> Rcpp::unary_call< T, RESULT_TYPE >::unary_call (Language *call_*) [inline]

Definition at line 182 of file Language.h.

6.786.2.2 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 183 of file Language.h.

6.786.2.3 template<typename T , typename RESULT_TYPE = SEXP> Rcpp::unary_call< T, RESULT_TYPE >::unary_call (Function *fun*) [inline]

Definition at line 184 of file Language.h.

6.786.3 Member Function Documentation

6.786.3.1 template<typename T , typename RESULT_TYPE = SEXP> RESULT_TYPE Rcpp::unary_call< T, RESULT_TYPE >::operator() (const T & *object*) [inline]

Definition at line 186 of file Language.h.

References Rcpp::fixed_call< RESULT_TYPE >::call.

6.786.4 Member Data Documentation

6.786.4.1 template<typename T , typename RESULT_TYPE = SEXP> Language Rcpp::unary_call< T, RESULT_TYPE >::call [private]

Definition at line 192 of file Language.h.

6.786.4.2 template<typename T , typename RESULT_TYPE = SEXP> Language::Proxy Rcpp::unary_call< T, RESULT_TYPE >::proxy [private]

Definition at line 193 of file Language.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.787 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

6.787.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::unary_minus< RTYPE, NA >

Definition at line 41 of file unary_minus.h.

6.787.2 Member Typedef Documentation

6.787.2.1 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.

6.787.2.2 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.

6.787.3 Member Function Documentation

6.787.3.1 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.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

6.788 Rcpp::sugar::unary_minus< CPLXSXP, false > Class Template Reference

`#include <unary_minus.h>`

Public Member Functions

- Rcomplex apply (Rcomplex x) const

6.788.1 Detailed Description

template<>
class Rcpp::sugar::unary_minus< CPLXSXP, false >

Definition at line 71 of file unary_minus.h.

6.788.2 Member Function Documentation

6.788.2.1 Rcomplex Rcpp::sugar::unary_minus< CPLXSXP, 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

6.789 Rcpp::sugar::unary_minus< CPLXSXP, NA > Class Template Reference

`#include <unary_minus.h>`

Public Member Functions

- Rcomplex apply (Rcomplex x) const

6.789.1 Detailed Description

template<bool NA>
class Rcpp::sugar::unary_minus< CPLXSXP, NA >

Definition at line 59 of file unary_minus.h.

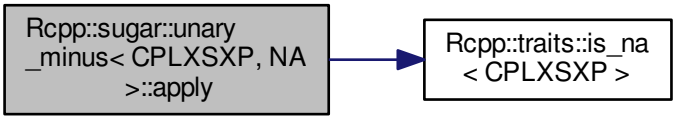
6.789.2 Member Function Documentation

6.789.2.1 template<bool NA> Rcomplex Rcpp::sugar::unary_minus< CPLXSXP, NA >::apply (Rcomplex *x*) const [inline]

Definition at line 61 of file unary_minus.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/unary_minus.h

6.790 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

6.790.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::unary_minus< RTYPE, false >

Definition at line 50 of file unary_minus.h.

6.790.2 Member Typedef Documentation

6.790.2.1 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.

6.790.2.2 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.

6.790.3 Member Function Documentation

6.790.3.1 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

6.791 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

6.791.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::unary_minus_result_type< RTYPE >

Definition at line 29 of file unary_minus.h.

6.791.2 Member Typedef Documentation

6.791.2.1 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.

6.791.3 Member Enumeration Documentation

6.791.3.1 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

6.792 Rcpp::sugar::unary_minus_result_type< LGLSXP > Struct Template Reference

`#include <unary_minus.h>`

Public Types

- enum { value = INTSXP }
- typedef traits::storage_type< INTSXP >::type type

6.792.1 Detailed Description

template<>
struct Rcpp::sugar::unary_minus_result_type< LGLSXP >

Definition at line 34 of file unary_minus.h.

6.792.2 Member Typedef Documentation

6.792.2.1 typedef traits::storage_type<INTSXP>::type Rcpp::sugar::unary_minus_result_type< LGLSXP >::type

Definition at line 35 of file unary_minus.h.

6.792.3 Member Enumeration Documentation

6.792.3.1 anonymous enum

Enumerator

value

Definition at line 36 of file unary_minus.h.

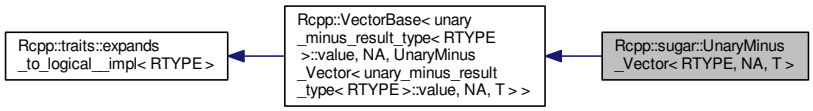
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

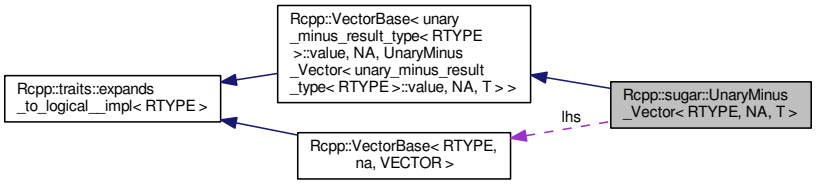
6.793 Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > Class Template Reference

`#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

6.793.1 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.

6.793.2 Member Typedef Documentation

6.793.2.1 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.

6.793.2.2 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.

6.793.2.3 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.

6.793.2.4 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.

6.793.3 Constructor & Destructor Documentation

6.793.3.1 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.

6.793.4 Member Function Documentation

6.793.4.1 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.

6.793.4.2 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.

6.793.5 Member Data Documentation

6.793.5.1 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.

6.793.5.2 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.

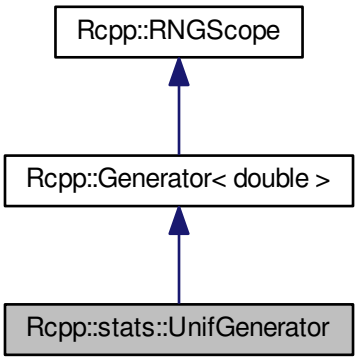
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

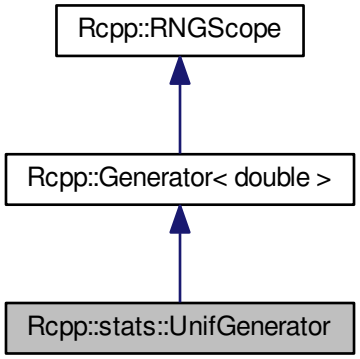
6.794 Rcpp::stats::UnifGenerator Class Reference

`#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

6.794.1 Detailed Description

Definition at line 28 of file runif.h.

6.794.2 Constructor & Destructor Documentation

6.794.2.1 Rcpp::stats::UnifGenerator::UnifGenerator (double *min_* = 0.0, double *max_* = 1.0) [inline]

Definition at line 31 of file runif.h.

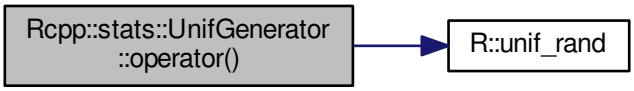
6.794.3 Member Function Documentation

6.794.3.1 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:



6.794.4 Member Data Documentation

6.794.4.1 double Rcpp::stats::UnifGenerator::diff [private]

Definition at line 42 of file runif.h.

Referenced by operator()().

6.794.4.2 double Rcpp::stats::UnifGenerator::min [private]

Definition at line 41 of file runif.h.

Referenced by operator()().

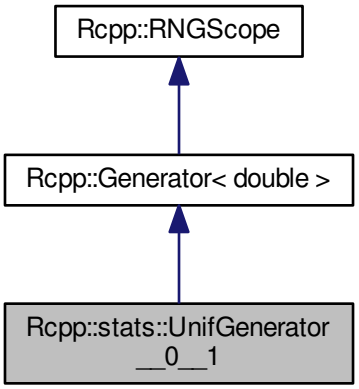
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/runif.h

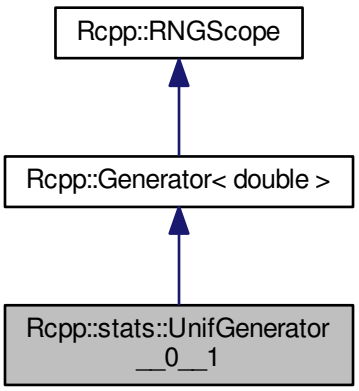
6.795 Rcpp::stats::UnifGenerator__0__1 Class Reference

`#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

6.795.1 Detailed Description

Definition at line 45 of file runif.h.

6.795.2 Constructor & Destructor Documentation

6.795.2.1 Rcpp::stats::UnifGenerator__0__1::UnifGenerator__0__1 ( ) [inline]

Definition at line 48 of file runif.h.

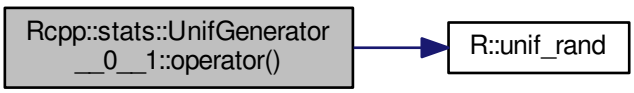
6.795.3 Member Function Documentation

6.795.3.1 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

6.796 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

6.796.1 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.

6.796.2 Member Typedef Documentation

6.796.2.1 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.

6.796.2.2 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.

6.796.2.3 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.

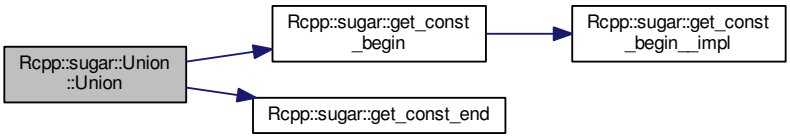
6.796.3 Constructor & Destructor Documentation

6.796.3.1 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:



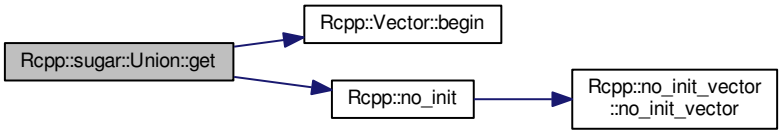
6.796.4 Member Function Documentation

6.796.4.1 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().

Here is the call graph for this function:



6.796.5 Member Data Documentation

6.796.5.1 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.

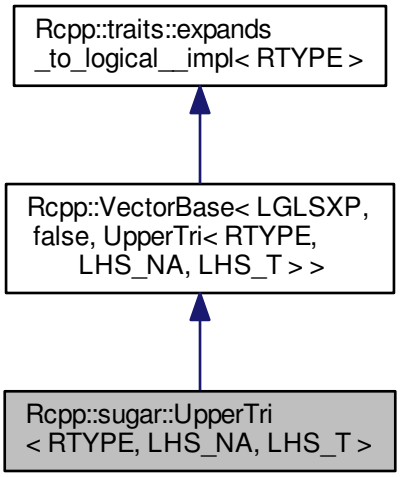
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

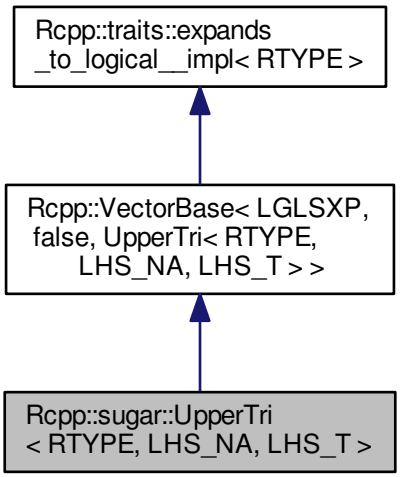
6.797 Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T > Class Template Reference

`#include <upper_tri.h>`

Inheritance diagram for Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >:



Collaboration diagram for Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >:



Public Types

- typedef Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE

Public Member Functions

- UpperTri (const LHS_TYPE &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)

Private Member Functions

- bool get_diag_true (int i, int j)
- bool get_diag_false (int i, int j)
- bool get (int i, int j)

Private Attributes

- int nr
- int nc
- Method getter

6.797.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >

Definition at line 29 of file upper_tri.h.

6.797.2 Member Typedef Documentation

6.797.2.1 template<int RTYPE, bool LHS_NA, typename LHS_T> typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::LHS_TYPE

Definition at line 35 of file upper_tri.h.

6.797.2.2 template<int RTYPE, bool LHS_NA, typename LHS_T> typedef bool(UpperTri::* Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::Method) (int, int) [private]

Definition at line 51 of file upper_tri.h.

6.797.3 Constructor & Destructor Documentation

6.797.3.1 template<int RTYPE, bool LHS_NA, typename LHS_T> Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::UpperTri (const LHS_TYPE & *lhs*, bool *diag*) [inline]

Definition at line 37 of file upper_tri.h.

6.797.4 Member Function Documentation

6.797.4.1 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::get (int *i*, int *j*) [inline], [private]

Definition at line 60 of file upper_tri.h.

6.797.4.2 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::get_diag_false (int *i*, int *j*) [inline], [private]

Definition at line 57 of file upper_tri.h.

6.797.4.3 template<int RTYPE, bool LHS_NA, typename LHS_T> bool Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::get_diag_true (int *i*, int *j*) [inline], [private]

Definition at line 54 of file upper_tri.h.

6.797.4.4 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::ncol ( ) const [inline]

Definition at line 47 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nc.

6.797.4.5 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nrow ( ) const [inline]

Definition at line 46 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nr.

6.797.4.6 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::operator() (int *i*, int *j*) const [inline]

Definition at line 41 of file upper_tri.h.

6.797.4.7 template<int RTYPE, bool LHS_NA, typename LHS_T> R_xlen_t Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]

Definition at line 45 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nc, and Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nr.

6.797.5 Member Data Documentation

6.797.5.1 template<int RTYPE, bool LHS_NA, typename LHS_T> Method Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::getter [private]

Definition at line 53 of file upper_tri.h.

6.797.5.2 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nc [private]

Definition at line 50 of file upper_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::ncol(), and Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::size().

6.797.5.3 template<int RTYPE, bool LHS_NA, typename LHS_T> int Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nr [private]

Definition at line 50 of file upper_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::nrow(), and Rcpp::sugar::UpperTri< RTYPE, LHS_NA, LHS_T >::size().

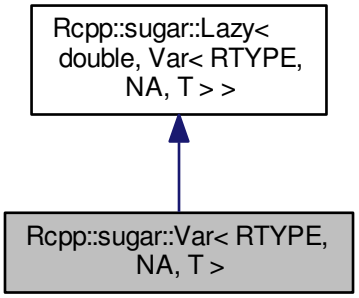
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/upper_tri.h

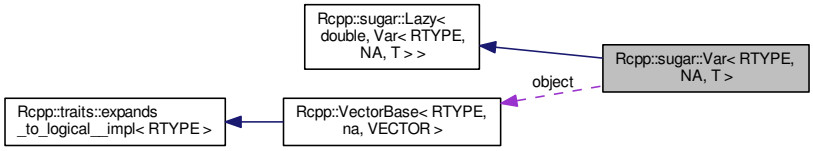
6.798 Rcpp::sugar::Var< RTYPE, NA, T > Class Template Reference

`#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

6.798.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Var< RTYPE, NA, T >

Definition at line 30 of file var.h.

6.798.2 Member Typedef Documentation

6.798.2.1 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.

6.798.3 Constructor & Destructor Documentation

6.798.3.1 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.

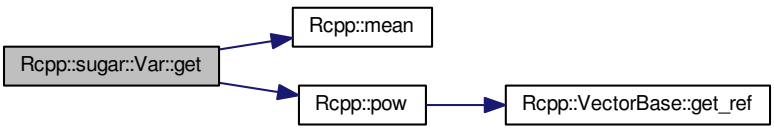
6.798.4 Member Function Documentation

6.798.4.1 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:



6.798.5 Member Data Documentation

6.798.5.1 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.

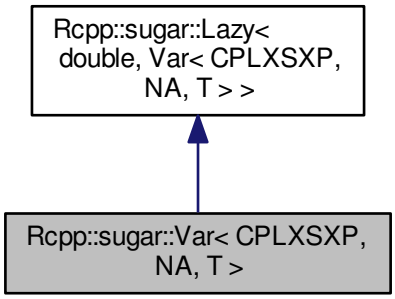
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/var.h

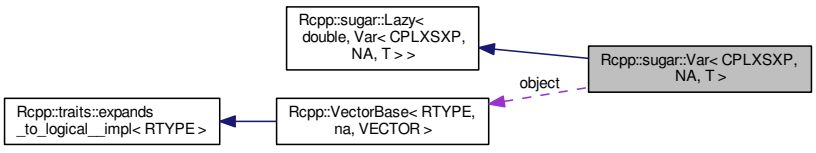
6.799 Rcpp::sugar::Var< CPLXSXP, NA, T > Class Template Reference

`#include <var.h>`

Inheritance diagram for Rcpp::sugar::Var< CPLXSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Var< CPLXSXP, NA, T >:



Public Types

- typedef Rcpp::VectorBase< CPLXSXP, NA, T > VEC_TYPE

Public Member Functions

- Var (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_TYPE & object

6.799.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Var< CPLXSXP, NA, T >

Definition at line 50 of file var.h.

6.799.2 Member Typedef Documentation

6.799.2.1 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.

6.799.3 Constructor & Destructor Documentation

6.799.3.1 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.

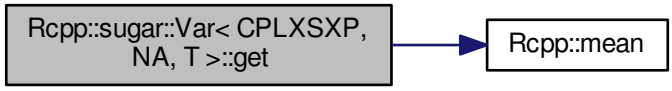
6.799.4 Member Function Documentation

6.799.4.1 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:



6.799.5 Member Data Documentation

6.799.5.1 template<bool NA, typename T > const VEC_TYPE& Rcpp::sugar::Var< CPLXSXP, NA, T >::object [private]

Definition at line 68 of file var.h.

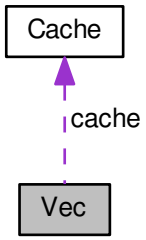
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/var.h

6.800 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

6.800.1 Detailed Description

Definition at line 16 of file convolve10_cpp.h.

6.800.2 Member Typedef Documentation

6.800.2.1 typedef double& Vec::proxy

Definition at line 18 of file convolve10_cpp.h.

6.800.2.2 typedef double& Vec::proxy

Definition at line 31 of file convolve9_cpp.cpp.

6.800.3 Constructor & Destructor Documentation

6.800.3.1 Vec::Vec (double * *data_*) [inline]

Definition at line 20 of file convolve10_cpp.h.

6.800.3.2 Vec::Vec (double * *data_*) [inline]

Definition at line 16 of file convolve8_cpp.cpp.

6.800.3.3 Vec::Vec (double * *data_*) [inline]

Definition at line 33 of file convolve9_cpp.cpp.

6.800.4 Member Function Documentation

6.800.4.1 double& Vec::operator[] (int *i*) [inline]

Definition at line 17 of file convolve8_cpp.cpp.

References data.

6.800.4.2 proxy Vec::operator[] (int *i*) [inline]

Definition at line 21 of file convolve10_cpp.h.

6.800.4.3 proxy Vec::operator[] (int *i*) const [inline]

Definition at line 22 of file convolve10_cpp.h.

6.800.4.4 proxy Vec::operator[] (int *i*) [inline]

Definition at line 34 of file convolve9_cpp.cpp.

6.800.4.5 proxy Vec::operator[] (int *i*) const [inline]

Definition at line 35 of file convolve9_cpp.cpp.

6.800.5 Member Data Documentation

6.800.5.1 Cache Vec::cache [private]

Definition at line 25 of file convolve10_cpp.h.

6.800.5.2 double* Vec::data [private]

Definition at line 20 of file convolve8_cpp.cpp.

Referenced by operator[ ]().

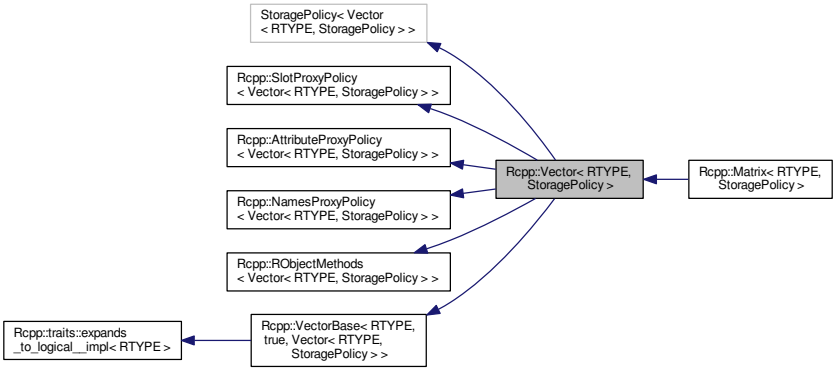
The documentation for this class was generated from the following files:

- inst/examples/ConvolveBenchmarks/convolve10_cpp.h
- inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
- inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp

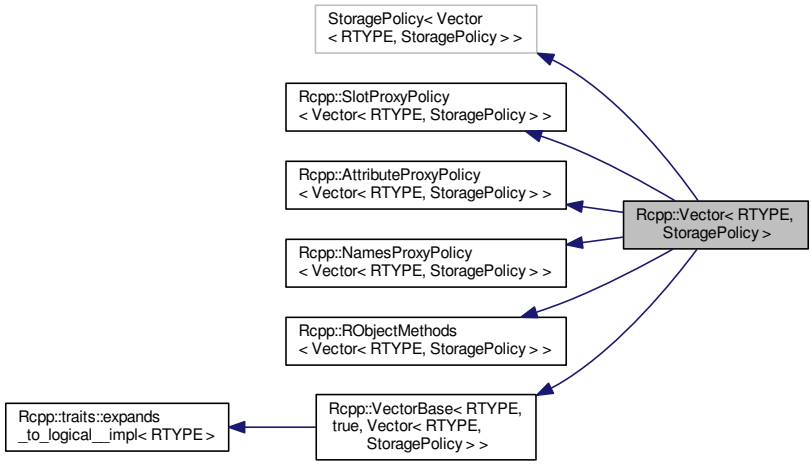
6.801 Rcpp::Vector< RTYPE, StoragePolicy > Class Template Reference

```
#include <00_forward_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 >::type Proxy
- typedef traits::r_vector_const_proxy< RTYPE >::type const_Proxy
- typedef traits::r_vector_name_proxy< RTYPE >::type NameProxy
- typedef traits::r_vector_proxy< RTYPE >::type value_type
- typedef traits::r_vector_iterator< RTYPE >::type iterator
- typedef traits::r_vector_const_iterator< RTYPE >::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 &siz, 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 &siz, 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 &siz, 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 &siz, 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

- 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< RH↩
  S_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, int 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)

6.801.1 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 26 of file 00_forward_Vector.h.

6.801.2 Member Typedef Documentation

6.801.2.1 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_const_iterator<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::const_iterator

Definition at line 48 of file Vector.h.

6.801.2.2 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::const_Proxy

Definition at line 44 of file Vector.h.

6.801.2.3 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 50 of file Vector.h.

6.801.2.4 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef internal::RangeIndexer<RTYPE,true,Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Indexer

Definition at line 528 of file Vector.h.

6.801.2.5 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::init_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::init_type

Definition at line 49 of file Vector.h.

6.801.2.6 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_iterator<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::iterator

Definition at line 47 of file Vector.h.

6.801.2.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_name_proxy<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::NameProxy

Definition at line 45 of file Vector.h.

6.801.2.8 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_proxy<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::Proxy

Definition at line 43 of file Vector.h.

6.801.2.9 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef StoragePolicy<Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Storage

Definition at line 40 of file Vector.h.

6.801.2.10 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::storage_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::stored_type

Definition at line 51 of file Vector.h.

6.801.2.11 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> typedef traits::r_vector_proxy<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::value_type

Definition at line 46 of file Vector.h.

6.801.3 Constructor & Destructor Documentation

6.801.3.1 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 57 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::create(), Rcpp::Vector< INTSXP >::import_transform(), and Rcpp::Vector< INTSXP >::Vector().

6.801.3.2 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 65 of file Vector.h.

6.801.3.3 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (SEXP *x*) [inline]

Definition at line 73 of file Vector.h.

6.801.3.4 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.

6.801.3.5 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const no_init_vector & *obj*) [inline], [explicit]

Definition at line 82 of file Vector.h.

6.801.3.6 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 87 of file Vector.h.

6.801.3.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const int & *size,* const stored_type & *u*) [inline]

Definition at line 94 of file Vector.h.

6.801.3.8 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const std::string & *st*) [inline]

Definition at line 101 of file Vector.h.

6.801.3.9 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const char * *st*) [inline]

Definition at line 107 of file Vector.h.

6.801.3.10 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 113 of file Vector.h.

6.801.3.11 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 122 of file Vector.h.

6.801.3.12 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const int & *size*) [inline]

Definition at line 128 of file Vector.h.

6.801.3.13 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::Vector (const Dimension & *dims*) [inline]

Definition at line 133 of file Vector.h.

6.801.3.14 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 145 of file Vector.h.

6.801.3.15 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 152 of file Vector.h.

6.801.3.16 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 162 of file Vector.h.

6.801.3.17 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 168 of file Vector.h.

6.801.3.18 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 176 of file Vector.h.

6.801.3.19 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 182 of file Vector.h.

6.801.3.20 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 191 of file Vector.h.

6.801.3.21 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 200 of file Vector.h.

6.801.3.22 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename InputIterator > Rcpp::Vector< RTYPE, StoragePolicy >::Vector (InputIterator *first,* InputIterator *last*) [inline]

Definition at line 209 of file Vector.h.

6.801.3.23 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 216 of file Vector.h.

6.801.3.24 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 224 of file Vector.h.

6.801.3.25 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 231 of file Vector.h.

6.801.4 Member Function Documentation

6.801.4.1 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 413 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::Vector().

6.801.4.2 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 1004 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::operator=().

6.801.4.3 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 1010 of file Vector.h.

6.801.4.4 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 989 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::assign_object().

6.801.4.5 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (const size_t & *i*) [inline]

Definition at line 334 of file Vector.h.

Referenced by Rcpp::attributes::SourceFileAttributesParser::parseSignature().

6.801.4.6 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (const size_t & *i*) const [inline]

Definition at line 337 of file Vector.h.

6.801.4.7 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( ) [inline]

Examples:

ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 319 of file Vector.h.

Referenced by Rcpp::ListOf< T >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::begin(), convolve10cpp(), convolve12cpp(), convolve4cpp(), convolve8cpp(), convolve9cpp(), Rcpp::Vector< INTSXP >::erase_range__impl(), Rcpp::Vector< INTSXP >::erase_single__impl(), fastLm(), Rcpp::Vector< INTSXP >::fill__dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), Rcpp::Vector< INTSXP >::fill_or_generate__impl(), Rcpp::sugar::Mean< RTYPE, 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::Mean< INTSXP, NA, T >::get(), Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::Vector< INTSXP >::import_expression(), Rcpp::Matrix< RTYPE, StoragePolicy >::import_matrix_expression(), Rcpp::Vector< INTSXP >::insert__impl(), Rcpp::Vector< INTSXP >::is_na(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Vector< INTSXP >::operator+=(), Rcpp::operator<<(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), Rcpp::Vector< INTSXP >::push_back__impl(), Rcpp::Vector< INTSXP >::push_back_name__impl(), Rcpp::Vector< INTSXP >::push_front__impl(), Rcpp::Vector< INTSXP >::push_front_name__impl(), stack_trace(), and Rcpp::Vector< INTSXP >::Vector().

6.801.4.8 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( ) const [inline]

Definition at line 321 of file Vector.h.

6.801.4.9 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 569 of file Vector.h.

6.801.4.10 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> static Vector Rcpp::Vector< RTYPE, StoragePolicy >::create ( ) [inline], [static]

Definition at line 1073 of file Vector.h.

Referenced by fastLm(), Rcpp::Module::get_function(), Rcpp::Module::invoke(), class_< Class >::invoke(), Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >(), Rcpp::shush_about_NA(), sourceCppContext(), and stack_trace().

6.801.4.11 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> int* Rcpp::Vector< RTYPE, StoragePolicy >::dims ( ) const [inline], [protected]

Definition at line 596 of file Vector.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::cols(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), Rcpp::Vector< INTSXP >::offset(), Rcpp::tranpose_impl(), and Rcpp::Vector< INTSXP >::Vector().

6.801.4.12 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( ) [inline]

Definition at line 320 of file Vector.h.

Referenced by Rcpp::ListOf< T >::end(), Rcpp::Matrix< RTYPE, StoragePolicy >::end(), Rcpp::Vector< INTSXP >::erase_range__impl(), Rcpp::Vector< INTSXP >::erase_single__impl(), Rcpp::Vector< INTSXP >::fill__dispatch(), Rcpp::Vector< INTSXP >::fill_or_generate__impl(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::Vector< INTSXP >::insert__impl(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::operator<<(), Rcpp::Vector< INTSXP >::push_back__impl(), Rcpp::Vector< INTSXP >::push_back_name__impl(), Rcpp::Vector< INTSXP >::push_front__impl(), Rcpp::Vector< INTSXP >::push_front_name__impl(), and Rcpp::Vector< INTSXP >::Vector().

6.801.4.13 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( ) const [inline]

Definition at line 322 of file Vector.h.

6.801.4.14 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (int *position*) [inline]

Definition at line 475 of file Vector.h.

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::from_list().

6.801.4.15 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (iterator *position*) [inline]

Definition at line 479 of file Vector.h.

6.801.4.16 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (int *first,* int *last*) [inline]

Definition at line 483 of file Vector.h.

6.801.4.17 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (iterator *first,* iterator *last*) [inline]

Definition at line 488 of file Vector.h.

6.801.4.18 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 946 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::erase().

6.801.4.19 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl (iterator *position*) [inline], [private]

Definition at line 907 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::erase().

6.801.4.20 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval ( ) const [inline]

Definition at line 1081 of file Vector.h.

6.801.4.21 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval (SEXP *env*) const [inline]

Definition at line 1085 of file Vector.h.

6.801.4.22 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename U > void Rcpp::Vector< RTYPE, StoragePolicy >::fill (const U & *u*) [inline]

Definition at line 315 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::fill_or_generate__impl(), and Rcpp::Vector< INTSXP >::Vector().

6.801.4.23 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 1057 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::fill().

6.801.4.24 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 1067 of file Vector.h.

6.801.4.25 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 1040 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::Vector().

6.801.4.26 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 1045 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::fill_or_generate().

6.801.4.27 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 1052 of file Vector.h.

6.801.4.28 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> int Rcpp::Vector< RTYPE, StoragePolicy >::findName (const std::string & *name*) const [inline]

Definition at line 580 of file Vector.h.

6.801.4.29 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> static stored_type Rcpp::Vector< RTYPE, StoragePolicy >::get_na ( ) [inline], [static]

Definition at line 250 of file Vector.h.

6.801.4.30 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 423 of file Vector.h.

6.801.4.31 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename T > void Rcpp::Vector< RTYPE, StoragePolicy >::import_expression (const T & *other,* int *n*) [inline], [private]

Definition at line 1034 of file Vector.h.

6.801.4.32 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 1018 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::Vector().

6.801.4.33 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 1027 of file Vector.h.

6.801.4.34 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 430 of file Vector.h.

6.801.4.35 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> void Rcpp::Vector< RTYPE, StoragePolicy >::init ( ) [inline], [protected]

Definition at line 600 of file Vector.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Vector< INTSXP >::Vector().

6.801.4.36 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 462 of file Vector.h.

Referenced by compileAttributes(), and Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::↩ Intersect().

6.801.4.37 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 469 of file Vector.h.

6.801.4.38 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 826 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::insert().

6.801.4.39 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 867 of file Vector.h.

6.801.4.40 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> static bool Rcpp::Vector< RTYPE, StoragePolicy >::is_na (stored_type *x*) [inline], [static]

Definition at line 253 of file Vector.h.

6.801.4.41 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 267 of file Vector.h.

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

6.801.4.42 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 281 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::at(), Rcpp::internal::simple_name_proxy< RTYPE >::get(), Rcpp::internal::string_name_proxy< RTYPE >::get(), Rcpp::internal::generic_name_proxy< RTYPE >::get(), Rcpp::Vector< INTSXP >::operator()(), Rcpp::internal::simple_name_proxy< RTYPE >::set(), Rcpp::internal::string_name_proxy< RTYPE >::set(), and Rcpp::internal::generic_name_proxy< RTYPE >::set().

6.801.4.43 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 296 of file Vector.h.

6.801.4.44 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 301 of file Vector.h.

6.801.4.45 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Rcpp::Vector< RTYPE, StoragePolicy >::operator RObject ( ) const [inline]

Definition at line 362 of file Vector.h.

6.801.4.46 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (const size_t & *i*) [inline]

Definition at line 327 of file Vector.h.

6.801.4.47 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (const size_t & *i*) const [inline]

Definition at line 330 of file Vector.h.

6.801.4.48 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 341 of file Vector.h.

6.801.4.49 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 344 of file Vector.h.

6.801.4.50 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (const std::string & *name*) [inline]

Definition at line 351 of file Vector.h.

6.801.4.51 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (const std::string & *name*) const [inline]

Definition at line 358 of file Vector.h.

6.801.4.52 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 535 of file Vector.h.

6.801.4.53 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 552 of file Vector.h.

6.801.4.54 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (const Vector< RTYPE, StoragePolicy > & *rhs*) [inline]

Definition at line 69 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::is_na().

6.801.4.55 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> template<typename T > Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (const T & *x*) [inline]

Definition at line 245 of file Vector.h.

6.801.4.56 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ] (R_xlen_t *i*) [inline]

Definition at line 324 of file Vector.h.

6.801.4.57 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ] (R_xlen_t *i*) const [inline]

Definition at line 325 of file Vector.h.

6.801.4.58 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ] (const std::string & *name*) [inline]

Definition at line 348 of file Vector.h.

6.801.4.59 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ] (const std::string & *name*) const [inline]

Definition at line 355 of file Vector.h.

6.801.4.60 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 369 of file Vector.h.

6.801.4.61 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 378 of file Vector.h.

6.801.4.62 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Indexer Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ] (const Range & *range*) [inline]

Definition at line 530 of file Vector.h.

6.801.4.63 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 435 of file Vector.h.

Referenced by Rcpp::attributes::isRoxygenCpp(), Rcpp::internal::simple_name_proxy< RTYPE >::set(), Rcpp::internal::string_name_proxy< RTYPE >::set(), and Rcpp::internal::generic_name_proxy< RTYPE >::set().

6.801.4.64 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 442 of file Vector.h.

6.801.4.65 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 607 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::push_back().

6.801.4.66 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 633 of file Vector.h.

6.801.4.67 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 658 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::push_back().

6.801.4.68 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 685 of file Vector.h.

6.801.4.69 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 449 of file Vector.h.

6.801.4.70 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 455 of file Vector.h.

6.801.4.71 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 713 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::push_front().

6.801.4.72 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 740 of file Vector.h.

6.801.4.73 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 767 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::push_front().

6.801.4.74 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 796 of file Vector.h.

6.801.4.75 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 497 of file Vector.h.

6.801.4.76 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 503 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::replace_element().

6.801.4.77 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 508 of file Vector.h.

6.801.4.78 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 513 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::replace_element__dispatch().

6.801.4.79 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 521 of file Vector.h.

6.801.4.80 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 274 of file Vector.h.

Referenced by Rcpp::Vector< INTSXP >::assign_sugar_expression(), convolve(), convolve10cpp(), convolve11cpp(), convolve12cpp(), convolve14cpp(), convolve3cpp(), convolve4cpp(), convolve5cpp(), convolve8cpp(), convolve9cpp(), Rcpp::Vector< INTSXP >::end(), Rcpp::Vector< INTSXP >::erase_range__impl(), Rcpp::Vector< INTSXP >::erase_single__impl(), fastLm(), Rcpp::Vector< INTSXP >::fill__dispatch(), Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Mean< CPLXSXP, NA, T >::get(), Rcpp::sugar::Mean< LGLSXP, NA, T >::get(), Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::Vector< INTSXP >::insert__impl(), lapplyCpp(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::sugar::na_omit_impl(), Rcpp::Vector< INTSXP >::offset(), Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector(), Rcpp::Vector< INTSXP >::operator+=(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=(), Rcpp::attributes::SourceFileAttributesParser::parseAttribute(), Rcpp::attributes::SourceFileAttributesParser::parseSignature(), Rcpp::Vector< INTSXP >::push_back__impl(), Rcpp::Vector< INTSXP >::push_back_name__impl(), Rcpp::Vector< INTSXP >::push_front__impl(), Rcpp::Vector< INTSXP >::push_front_name__impl(), Rcpp::ListOf< T >::size(), Rcpp::Vector< INTSXP >::sort(), and Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser().

6.801.4.81 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> Vector& Rcpp::Vector< RTYPE, StoragePolicy >::sort (bool *decreasing* = false) [inline]

Definition at line 385 of file Vector.h.

6.801.4.82 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> void Rcpp::Vector< RTYPE, StoragePolicy >::update (SEXP) [inline]

Definition at line 492 of file Vector.h.

6.801.5 Member Data Documentation

6.801.5.1 template<int RTYPE, template< class > class StoragePolicy = PreserveStorage> traits::r_vector_cache_type<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy >::cache

Definition at line 42 of file Vector.h.

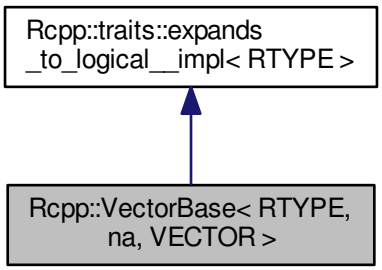
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_Vector.h
- inst/include/Rcpp/vector/Vector.h

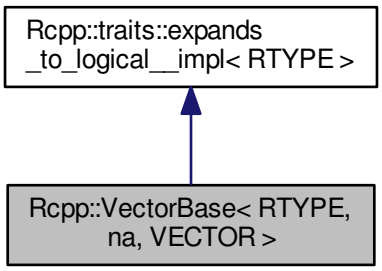
6.802 Rcpp::VectorBase< RTYPE, na, VECTOR > Class Template Reference

`#include <VectorBase.h>`

Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >:



Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >:



Classes

- struct can_have_na
- class iterator
- 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 iterator 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
- iterator begin () const
- iterator end () const

6.802.1 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.

6.802.2 Member Typedef Documentation

6.802.2.1 template<int RTYPE, bool na, typename VECTOR> typedef iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iterator

Definition at line 138 of file VectorBase.h.

6.802.2.2 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.

6.802.2.3 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.

6.802.3 Member Function Documentation

6.802.3.1 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::begin ( ) const [inline]

Definition at line 140 of file VectorBase.h.

6.802.3.2 template<int RTYPE, bool na, typename VECTOR> iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::end ( ) const [inline]

Definition at line 141 of file VectorBase.h.

6.802.3.3 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< INTSXP >::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< INTSXP >::operator+=(), Rcpp::MatrixColumn< RTYPE >::operator=(), Rcpp::MatrixRow< RTYPE >::operator=(), Rcpp::pmax(), Rcpp::pmin(), Rcpp::pow(), Rcpp::range(), Rcpp::setdiff(), Rcpp::setequal(), Rcpp::table(), and Rcpp::union_().

6.802.3.4 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.

6.802.3.5 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.

6.802.3.6 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.

Referenced by Rcpp::collapse(), Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, false, CO↩ND_T > >::end(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMax< RTYPE, NA, T >↩::get(), Rcpp::sugar::WhichMax< RTYPE, false, T >::get(), Rcpp::sugar::WhichMin< RTYPE, false, T >::get(), Rcpp↩::Vector< INTSXP >::import_sugar_expression(), Rcpp::seq_along(), Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size(), Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size(), Rcpp::sugar::SugarMath_1< NA, RESULT_TYP↩E, U1, T1, FunPtr >::size(), Rcpp::stats::D0< RTYPE, NA, T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::size(), Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, false, COND_T > >::size(), Rcpp↩::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size(), Rcpp::sugar::IfElse< RTYPE, COND_NA, CON↩D_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size(), Rcpp↩::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size(), Rcpp::stats::D1< RTYPE, NA, T >::size(), Rcpp::sugar::Sugar↩Math_1< NA, RESULT_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size(), Rcpp↩::stats::D2< RTYPE, NA, T >::size(), Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size(), Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, L↩HS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::size(), Rcpp::stats::D3< RTYPE, NA, T >::size(), Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size(), Rcpp::stats::P0< RTY↩PE, NA, T >::size(), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size(), Rcpp::stats::P1< RTYPE, NA, T >::size(), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size(), Rcpp::stats::P2< RTYPE, NA, T >::size(), Rcpp::stats::P3< RTYPE, NA, T >::size(), Rcpp::sugar↩::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::size(), Rcpp::stats::Q0< RTYPE, NA, T >::size(), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::size(), Rcpp::stats::Q1< RTYPE, NA, T >::size(), Rcpp::stats::Q2< RTYPE, NA, T >::size(), Rcpp::sugar::IfElse_Primitive_Primitive< RTY↩PE, COND_NA, COND_T >::size(), Rcpp::stats::Q3< RTYPE, NA, T >::size(), and Rcpp::sugar::IfElse_Primitive_↩Primitive< RTYPE, false, COND_T >::size().

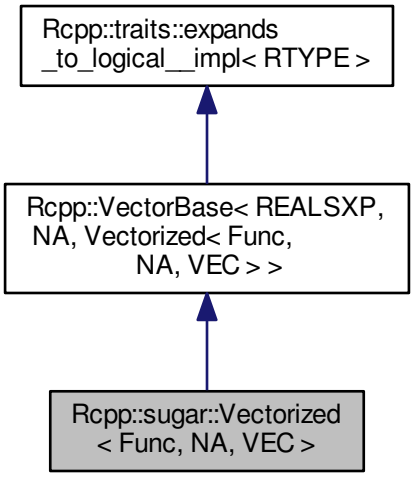
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

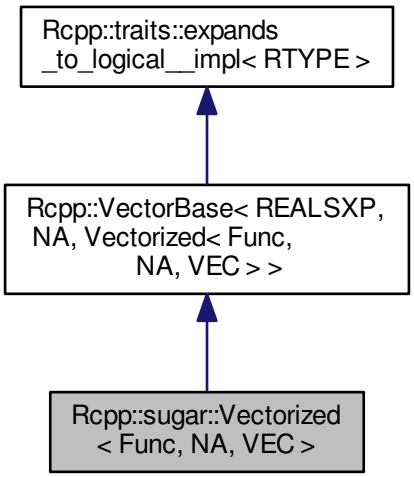
6.803 Rcpp::sugar::Vectorized< Func, NA, VEC > Class Template Reference

`#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

6.803.1 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.

6.803.2 Member Typedef Documentation

6.803.2.1 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.

6.803.2.2 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.

6.803.3 Constructor & Destructor Documentation

6.803.3.1 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.

6.803.4 Member Function Documentation

6.803.4.1 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.

6.803.4.2 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.

6.803.5 Member Data Documentation

6.803.5.1 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.

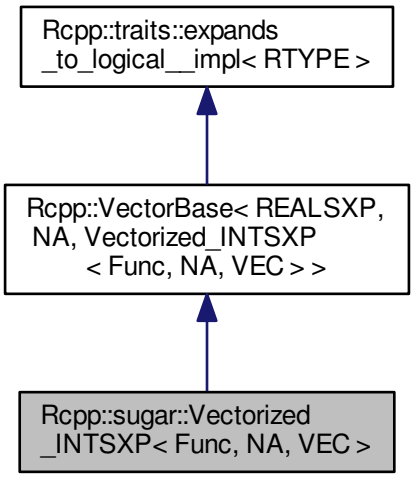
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized_Math.h

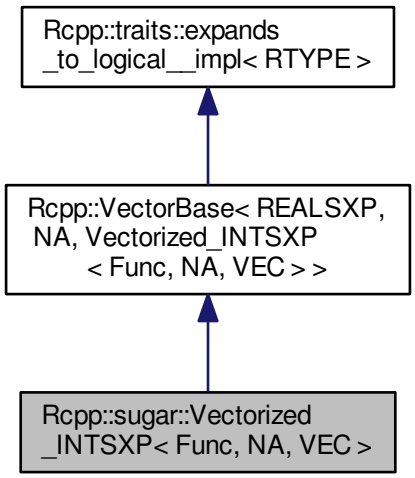
6.804 Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC > Class Template Reference

`#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

6.804.1 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.

6.804.2 Member Typedef Documentation

6.804.2.1 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.

6.804.2.2 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.

6.804.3 Constructor & Destructor Documentation

6.804.3.1 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.

6.804.4 Member Function Documentation

6.804.4.1 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.

6.804.4.2 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.

6.804.5 Member Data Documentation

6.804.5.1 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.

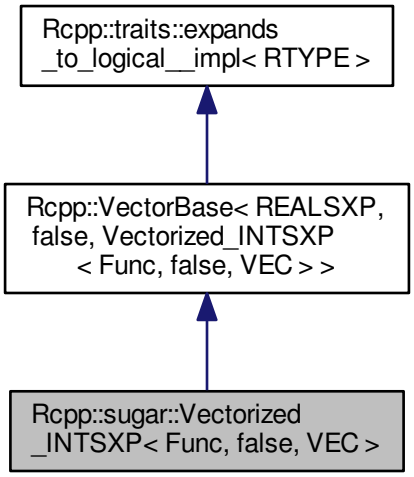
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized_Math.h

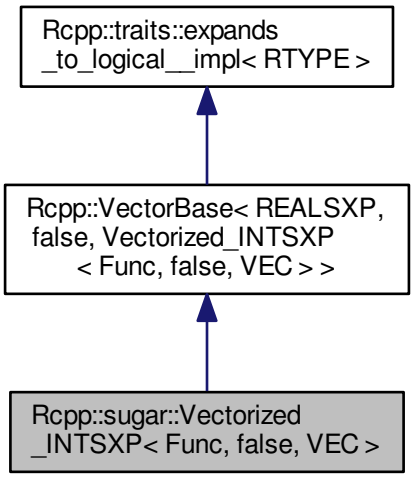
6.805 Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC > Class Template Reference

`#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

6.805.1 Detailed Description

template<DDFun Func, typename VEC>
class Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >

Definition at line 64 of file Vectorized_Math.h.

6.805.2 Member Typedef Documentation

6.805.2.1 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.

6.805.2.2 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.

6.805.3 Constructor & Destructor Documentation

6.805.3.1 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.

6.805.4 Member Function Documentation

6.805.4.1 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.

6.805.4.2 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.

6.805.5 Member Data Documentation

6.805.5.1 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

6.806 Rcpp::void_type Struct Reference

`#include <get_return_type.h>`

6.806.1 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

6.807 Rcpp::traits::void_wrap_tag Struct Reference

`#include <module_wrap_traits.h>`

6.807.1 Detailed Description

Definition at line 30 of file module_wrap_traits.h.

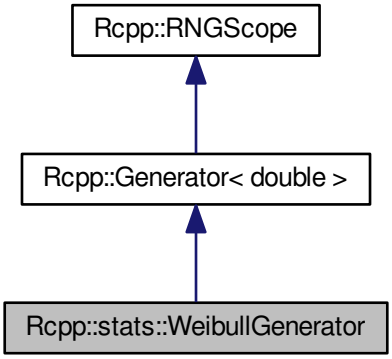
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

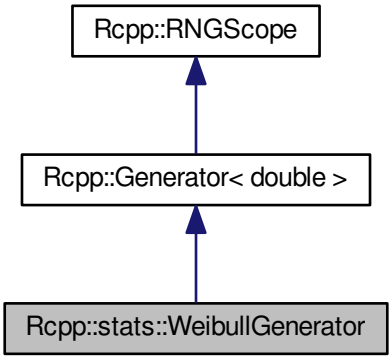
6.808 Rcpp::stats::WeibullGenerator Class Reference

`#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

6.808.1 Detailed Description

Definition at line 28 of file rweibull.h.

6.808.2 Constructor & Destructor Documentation

6.808.2.1 Rcpp::stats::WeibullGenerator::WeibullGenerator (double *shape_*, double *scale_*) [inline]

Definition at line 31 of file rweibull.h.

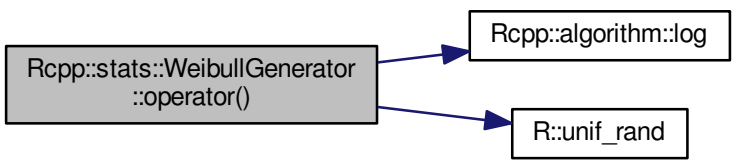
6.808.3 Member Function Documentation

6.808.3.1 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:



6.808.4 Member Data Documentation

6.808.4.1 double Rcpp::stats::WeibullGenerator::scale [private]

Definition at line 39 of file rweibull.h.

Referenced by operator()().

6.808.4.2 double Rcpp::stats::WeibullGenerator::shape_inv [private]

Definition at line 39 of file rweibull.h.

Referenced by operator()(), and Rcpp::stats::WeibullGenerator__scale1::operator()().

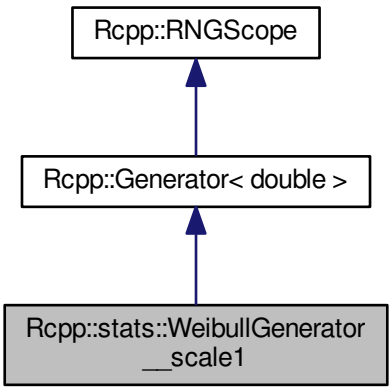
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rweibull.h

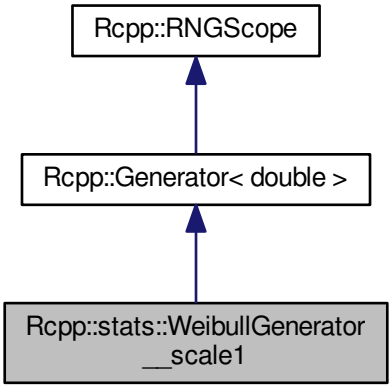
6.809 Rcpp::stats::WeibullGenerator__scale1 Class Reference

```
#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

6.809.1 Detailed Description

Definition at line 42 of file rweibull.h.

6.809.2 Constructor & Destructor Documentation

6.809.2.1 Rcpp::stats::WeibullGenerator__scale1::WeibullGenerator__scale1 (double *shape_*) [inline]

Definition at line 45 of file rweibull.h.

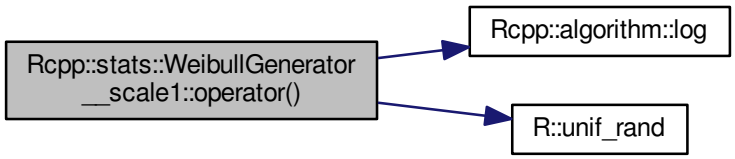
6.809.3 Member Function Documentation

6.809.3.1 double Rcpp::stats::WeibullGenerator__scale1::operator() ( ) const [inline]

Definition at line 48 of file rweibull.h.

References Rcpp::algorithm::log(), Rcpp::stats::WeibullGenerator::shape_inv, and R::unif_rand().

Here is the call graph for this function:



6.809.4 Member Data Documentation

6.809.4.1 double Rcpp::stats::WeibullGenerator__scale1::shape_inv [private]

Definition at line 53 of file rweibull.h.

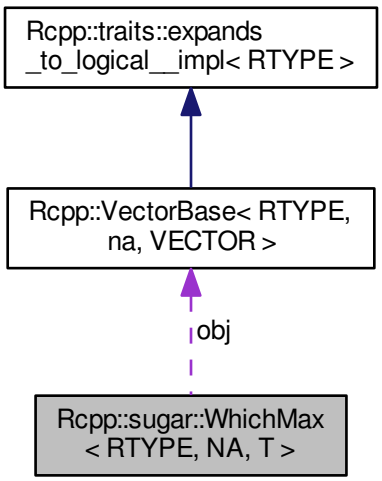
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rweibull.h

6.810 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

6.810.1 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.

6.810.2 Member Typedef Documentation

6.810.2.1 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.

6.810.2.2 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.

6.810.3 Constructor & Destructor Documentation

6.810.3.1 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.

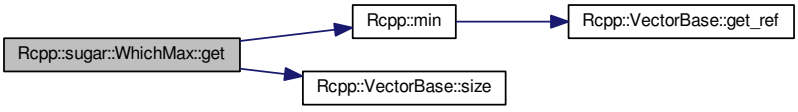
6.810.4 Member Function Documentation

6.810.4.1 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, VEC↩ TOR >::size().

Here is the call graph for this function:



6.810.5 Member Data Documentation

6.810.5.1 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().

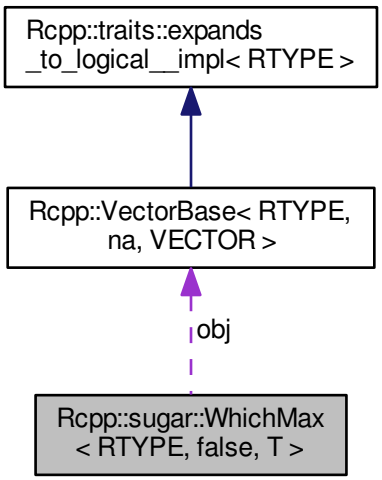
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_max.h

6.811 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

6.811.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::WhichMax< RTYPE, false, T >

Definition at line 58 of file which_max.h.

6.811.2 Member Typedef Documentation

6.811.2.1 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.

6.811.2.2 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.

6.811.3 Constructor & Destructor Documentation

6.811.3.1 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.

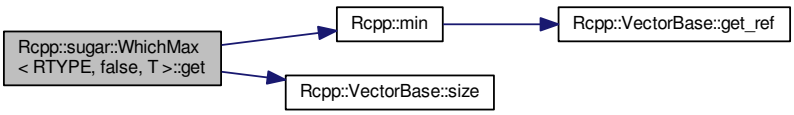
6.811.4 Member Function Documentation

6.811.4.1 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, VEC↩ TOR >::size().

Here is the call graph for this function:



6.811.5 Member Data Documentation

6.811.5.1 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.

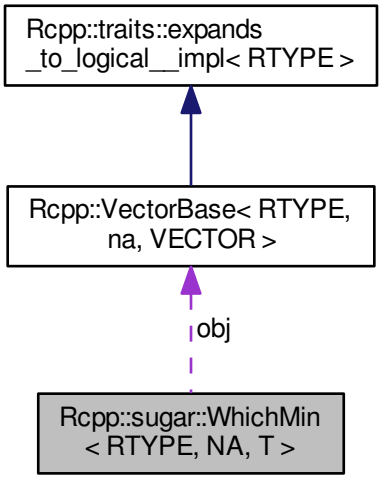
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_max.h

6.812 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

6.812.1 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.

6.812.2 Member Typedef Documentation

6.812.2.1 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.

6.812.2.2 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.

6.812.3 Constructor & Destructor Documentation

6.812.3.1 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.

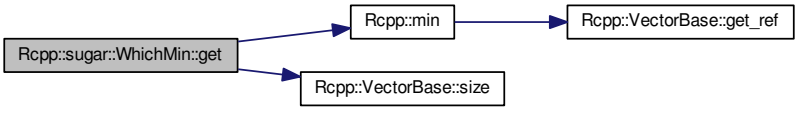
6.812.4 Member Function Documentation

6.812.4.1 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, VEC↩ TOR >::size().

Here is the call graph for this function:



6.812.5 Member Data Documentation

6.812.5.1 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().

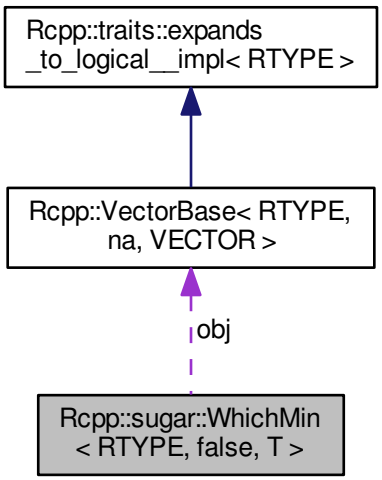
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_min.h

6.813 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

6.813.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::WhichMin< RTYPE, false, T >

Definition at line 58 of file which_min.h.

6.813.2 Member Typedef Documentation

6.813.2.1 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.

6.813.2.2 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.

6.813.3 Constructor & Destructor Documentation

6.813.3.1 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.

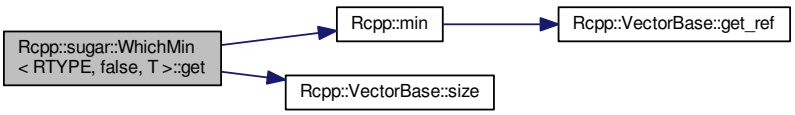
6.813.4 Member Function Documentation

6.813.4.1 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, VEC↩ TOR >::size().

Here is the call graph for this function:



6.813.5 Member Data Documentation

6.813.5.1 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.

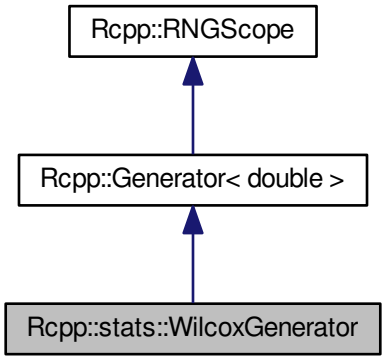
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_min.h

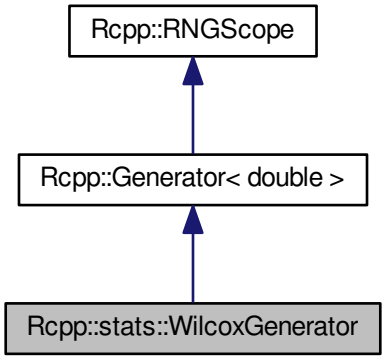
6.814 Rcpp::stats::WilcoxGenerator Class Reference

```
#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

6.814.1 Detailed Description

Definition at line 28 of file rwilcox.h.

6.814.2 Constructor & Destructor Documentation

6.814.2.1 Rcpp::stats::WilcoxGenerator::WilcoxGenerator (double *mm_*, double *nn_*) [inline]

Definition at line 30 of file rwilcox.h.

6.814.3 Member Function Documentation

6.814.3.1 double Rcpp::stats::WilcoxGenerator::operator() ( ) const [inline]

Definition at line 31 of file rwilcox.h.

References mm, and nn.

6.814.4 Member Data Documentation

6.814.4.1 double Rcpp::stats::WilcoxGenerator::mm [private]

Definition at line 33 of file rwilcox.h.

Referenced by operator()().

6.814.4.2 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

6.815 Rcpp::traits::wrap_type_char_array Struct Reference

```
#include <wrap_type_traits.h>
```

6.815.1 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

6.816 Rcpp::traits::wrap_type_enum_tag Struct Reference

`#include <wrap_type_traits.h>`

6.816.1 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

6.817 Rcpp::traits::wrap_type_module_object_pointer_tag Struct Reference

`#include <wrap_type_traits.h>`

6.817.1 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

6.818 Rcpp::traits::wrap_type_module_object_tag Struct Reference

`#include <wrap_type_traits.h>`

6.818.1 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

6.819 Rcpp::traits::wrap_type_primitive_tag Struct Reference

`#include <wrap_type_traits.h>`

6.819.1 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

6.820 Rcpp::traits::wrap_type_traits< T > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_unknown_tag wrap_category

6.820.1 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.

6.820.2 Member Typedef Documentation

6.820.2.1 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

6.821 Rcpp::traits::wrap_type_traits< bool > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.821.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< bool >

Definition at line 82 of file wrap_type_traits.h.

6.821.2 Member Typedef Documentation

6.821.2.1 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

6.822 Rcpp::traits::wrap_type_traits< char > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.822.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< char >

Definition at line 86 of file wrap_type_traits.h.

6.822.2 Member Typedef Documentation

6.822.2.1 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

6.823 Rcpp::traits::wrap_type_traits< char[N]> Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_char_array wrap_category

6.823.1 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.

6.823.2 Member Typedef Documentation

6.823.2.1 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

6.824 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

6.824.1 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.

6.824.2 Member Typedef Documentation

6.824.2.1 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

6.825 Rcpp::traits::wrap_type_traits< const int > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.825.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< const int >

Definition at line 76 of file wrap_type_traits.h.

6.825.2 Member Typedef Documentation

6.825.2.1 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

6.826 Rcpp::traits::wrap_type_traits< double > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.826.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< double >

Definition at line 78 of file wrap_type_traits.h.

6.826.2 Member Typedef Documentation

6.826.2.1 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

6.827 Rcpp::traits::wrap_type_traits< float > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.827.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< float >

Definition at line 89 of file wrap_type_traits.h.

6.827.2 Member Typedef Documentation

6.827.2.1 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

6.828 Rcpp::traits::wrap_type_traits< int > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.828.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< int >

Total specialization for primitive types

Definition at line 75 of file wrap_type_traits.h.

6.828.2 Member Typedef Documentation

6.828.2.1 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

6.829 Rcpp::traits::wrap_type_traits< long > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.829.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< long >

Definition at line 93 of file wrap_type_traits.h.

6.829.2 Member Typedef Documentation

6.829.2.1 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

6.830 Rcpp::traits::wrap_type_traits< long double > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.830.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< long double >

Definition at line 96 of file wrap_type_traits.h.

6.830.2 Member Typedef Documentation

6.830.2.1 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

6.831 Rcpp::traits::wrap_type_traits< Rbyte > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.831.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< Rbyte >

Definition at line 79 of file wrap_type_traits.h.

6.831.2 Member Typedef Documentation

6.831.2.1 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

6.832 Rcpp::traits::wrap_type_traits< Rcomplex > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.832.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< Rcomplex >

Definition at line 80 of file wrap_type_traits.h.

6.832.2 Member Typedef Documentation

6.832.2.1 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

6.833 Rcpp::traits::wrap_type_traits< Rcpp::Date > Struct Template Reference

`#include <Date.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.833.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< Rcpp::Date >

Definition at line 28 of file Date.h.

6.833.2 Member Typedef Documentation

6.833.2.1 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

6.834 Rcpp::traits::wrap_type_traits< Rcpp::Datetime > Struct Template Reference

`#include <Datetime.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.834.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< Rcpp::Datetime >

Definition at line 28 of file Datetime.h.

6.834.2 Member Typedef Documentation

6.834.2.1 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

6.835 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

6.835.1 Detailed Description

template<typename T>
struct Rcpp::traits::wrap_type_traits< Rcpp::object< T > >

Definition at line 101 of file wrap_type_traits.h.

6.835.2 Member Typedef Documentation

6.835.2.1 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

6.836 Rcpp::traits::wrap_type_traits< Rcpp::String > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.836.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< Rcpp::String >

Definition at line 85 of file wrap_type_traits.h.

6.836.2 Member Typedef Documentation

6.836.2.1 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

6.837 Rcpp::traits::wrap_type_traits< short > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.837.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< short >

Definition at line 98 of file wrap_type_traits.h.

6.837.2 Member Typedef Documentation

6.837.2.1 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

6.838 Rcpp::traits::wrap_type_traits< std::complex< double > > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.838.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< std::complex< double > >

Definition at line 91 of file wrap_type_traits.h.

6.838.2 Member Typedef Documentation

6.838.2.1 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

6.839 Rcpp::traits::wrap_type_traits< std::complex< float > > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.839.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< std::complex< float > >

Definition at line 90 of file wrap_type_traits.h.

6.839.2 Member Typedef Documentation

6.839.2.1 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

6.840 Rcpp::traits::wrap_type_traits< std::string > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.840.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< std::string >

Definition at line 83 of file wrap_type_traits.h.

6.840.2 Member Typedef Documentation

6.840.2.1 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

6.841 Rcpp::traits::wrap_type_traits< std::wstring > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.841.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< std::wstring >

Definition at line 84 of file wrap_type_traits.h.

6.841.2 Member Typedef Documentation

6.841.2.1 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

6.842 Rcpp::traits::wrap_type_traits< unsigned int > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.842.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< unsigned int >

Definition at line 81 of file wrap_type_traits.h.

6.842.2 Member Typedef Documentation

6.842.2.1 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

6.843 Rcpp::traits::wrap_type_traits< unsigned long > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.843.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< unsigned long >

Definition at line 94 of file wrap_type_traits.h.

6.843.2 Member Typedef Documentation

6.843.2.1 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

6.844 Rcpp::traits::wrap_type_traits< unsigned short > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.844.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< unsigned short >

Definition at line 99 of file wrap_type_traits.h.

6.844.2 Member Typedef Documentation

6.844.2.1 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

6.845 Rcpp::traits::wrap_type_traits< wchar_t > Struct Template Reference

`#include <wrap_type_traits.h>`

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.845.1 Detailed Description

template<>
struct Rcpp::traits::wrap_type_traits< wchar_t >

Definition at line 87 of file wrap_type_traits.h.

6.845.2 Member Typedef Documentation

6.845.2.1 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

6.846 Rcpp::traits::wrap_type_unknown_tag Struct Reference

`#include <wrap_type_traits.h>`

6.846.1 Detailed Description

unknown. Implicit conversion to SEXP will be tried.

Definition at line 37 of file wrap_type_traits.h.

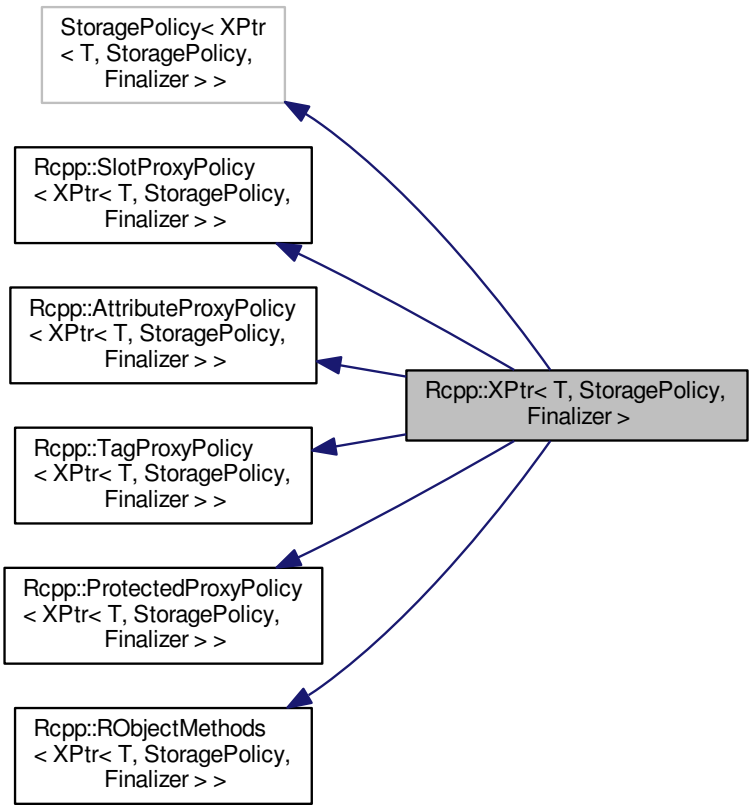
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

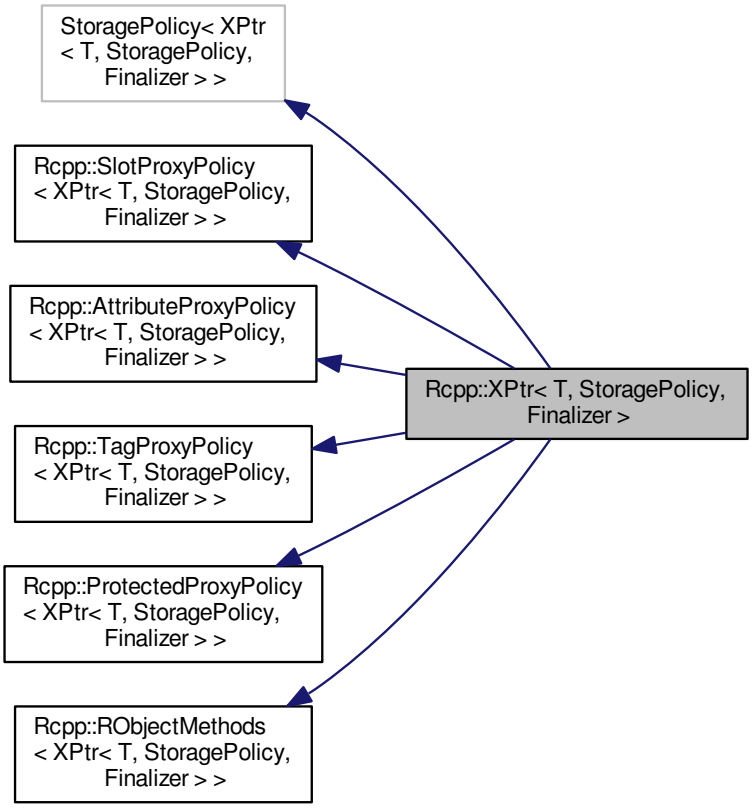
6.847 Rcpp::XPtr< T, StoragePolicy, Finalizer > Class Template Reference

`#include <XPtr.h>`

Inheritance diagram for Rcpp::XPtr< T, StoragePolicy, Finalizer >:



Collaboration diagram for Rcpp::XPtr< T, StoragePolicy, Finalizer >:



Public Types

- typedef StoragePolicy< XPtr > Storage
- typedef void(* unspecified_bool_type) ()

Public Member Functions

- XPtr (SEXP x, SEXP tag=R_NilValue, SEXP prot=R_NilValue)
- 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 ()

6.847.1 Detailed Description

template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> class Rcpp::XPtr< T, StoragePolicy, Finalizer >

Definition at line 48 of file XPtr.h.

6.847.2 Member Typedef Documentation

6.847.2.1 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> typedef StoragePolicy<XPtr> Rcpp::XPtr< T, StoragePolicy, Finalizer >::Storage

Definition at line 58 of file XPtr.h.

6.847.2.2 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> typedef void(* Rcpp::XPtr< T, StoragePolicy, Finalizer >::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

Definition at line 113 of file XPtr.h.

6.847.3 Constructor & Destructor Documentation

6.847.3.1 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> Rcpp::XPtr< T, StoragePolicy, Finalizer >::XPtr (SEXP *x*, SEXP *tag* = R_NilValue, SEXP *prot* = R_NilValue) [inline], [explicit]

constructs a XPtr wrapping the external pointer (EXTPTRSXP SEXP)

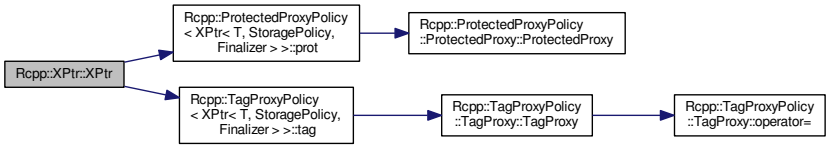
Parameters

xp	external pointer to wrap

Definition at line 65 of file XPtr.h.

References Rcpp::ProtectedProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::prot(), and Rcpp::TagProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::tag().

Here is the call graph for this function:



6.847.3.2 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> Rcpp::XPtr< T, StoragePolicy, Finalizer >::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.

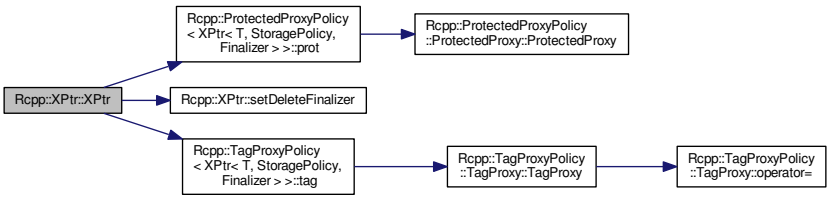
Parameters

p	dumb pointer to some object
set_delete_finalizer	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 84 of file XPtr.h.

References Rcpp::ProtectedProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::prot(), RCPP_DEBUG_2, Rcpp::XPtr< T, StoragePolicy, Finalizer >::setDeleteFinalizer(), and Rcpp::TagProxyPolicy< XPtr< T, StoragePolicy, Finalizer > >::tag().

Here is the call graph for this function:



6.847.3.3 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> Rcpp::XPtr< T, StoragePolicy, Finalizer >::XPtr (const XPtr< T, StoragePolicy, Finalizer > & *other*) [inline]

Definition at line 92 of file XPtr.h.

6.847.4 Member Function Documentation

6.847.4.1 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> T* Rcpp::XPtr< T, StoragePolicy, Finalizer >::checked_get ( ) const [inline]

Access underlying pointer throwing an exception if the ptr is NULL

Definition at line 127 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator T *(), Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator*(), and Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator->().

6.847.4.2 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> T* Rcpp::XPtr< T, StoragePolicy, Finalizer >::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 105 of file XPtr.h.

6.847.4.3 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator T * ( ) [inline]

Definition at line 177 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer >::checked_get().

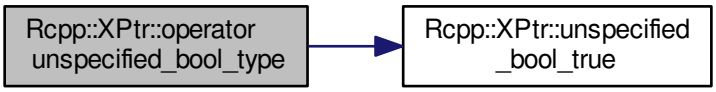
Here is the call graph for this function:

6.847.4.4 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator unspecified_bool_type ( ) const [inline]

Definition at line 115 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer >::unspecified_bool_true().

Here is the call graph for this function:



6.847.4.5 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> bool Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator! ( ) const [inline]

Definition at line 119 of file XPtr.h.

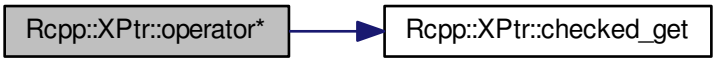
6.847.4.6 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> T& Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator* ( ) const [inline]

Returns a reference to the object wrapped. This allows this object to look and feel like a dumb pointer to T

Definition at line 138 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer >::checked_get().

Here is the call graph for this function:



6.847.4.7 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> T* Rcpp::XPtr< T, StoragePolicy, Finalizer >::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 146 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer >::checked_get().

Here is the call graph for this function:

6.847.4.8 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> XPtr& Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator= (const XPtr< T, StoragePolicy, Finalizer > & *other*) [inline]

Definition at line 96 of file XPtr.h.

6.847.4.9 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> void Rcpp::XPtr< T, StoragePolicy, Finalizer >::release ( ) [inline]

Release the external pointer (if any) immediately. This will cause the pointer to be deleted and it's 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 163 of file XPtr.h.

6.847.4.10 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> void Rcpp::XPtr< T, StoragePolicy, Finalizer >::setDeleteFinalizer ( ) [inline]

Definition at line 150 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer >::XPtr().

6.847.4.11 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> static void Rcpp::XPtr< T, StoragePolicy, Finalizer >::unspecified_bool_true ( ) [inline], [static]

Definition at line 114 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer >::operator unspecified_bool_type().

6.847.4.12 template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>> void Rcpp::XPtr< T, StoragePolicy, Finalizer >::update (SEXP) [inline]

Definition at line 181 of file XPtr.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/XPtr.h

Chapter 7

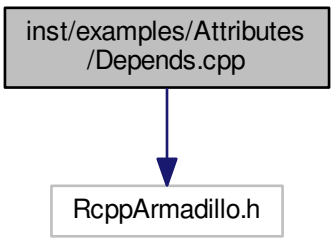
File Documentation

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)

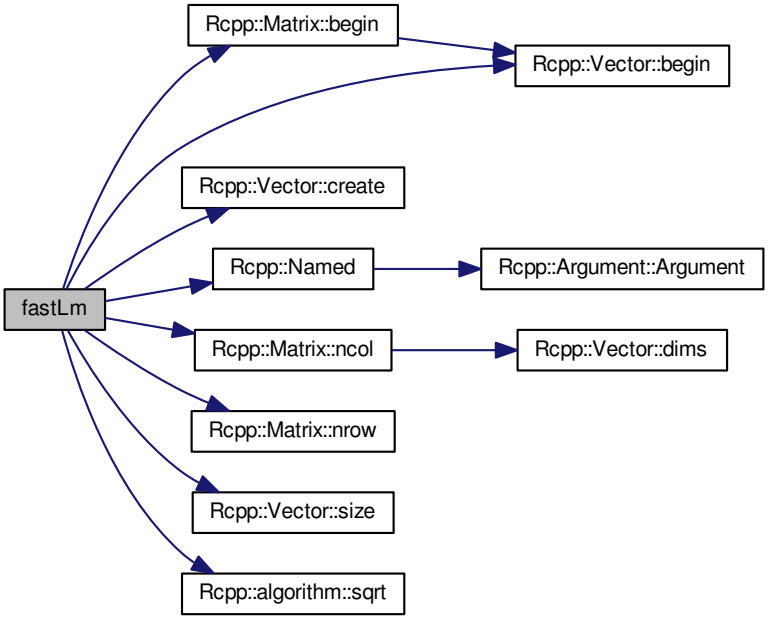
7.2.1 Function Documentation

7.2.1.1 List fastLm (NumericVector *yr*, NumericMatrix *Xr*)

Definition at line 9 of file Depends.cpp.

References Rcpp::Matrix< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::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 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 int fibonacci (const int *x*)

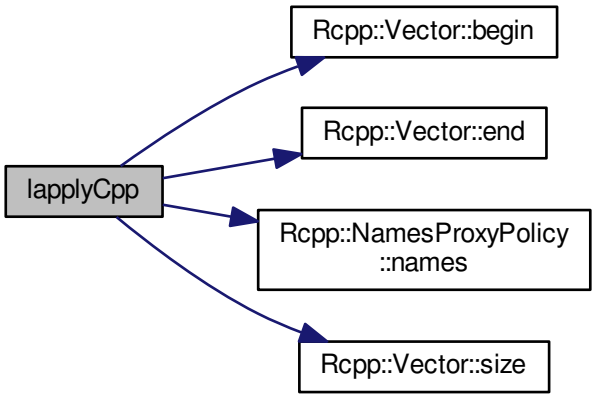
Definition at line 7 of file Export.cpp.

7.3.1.3 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:

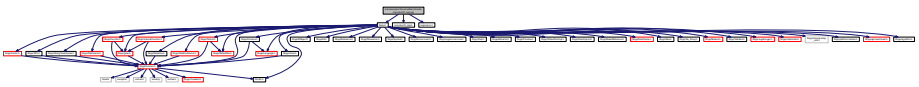


7.4 inst/examples/Attributes/sourceCpp.R File Reference

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)

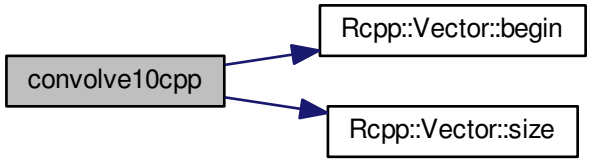
7.5.1 Function Documentation

7.5.1.1 RcppExport SEXP convolve10cpp (SEXP *a*, SEXP *b*)

Definition at line 17 of file convolve10_cpp.cpp.

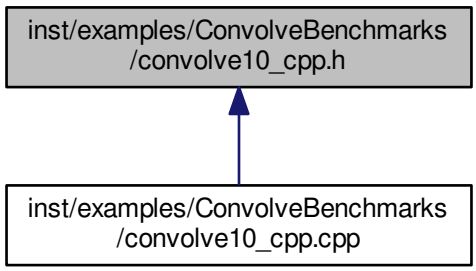
References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), LOOPMACRO_CPP, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

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)

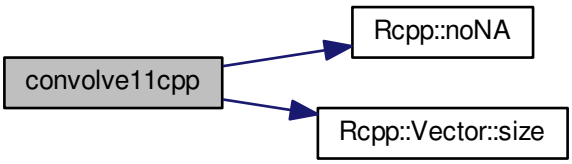
7.7.1 Function Documentation

7.7.1.1 RcppExport SEXP convolve11cpp (SEXP *a*, SEXP *b*)

Definition at line 10 of file convolve11_cpp.cpp.

References LOOPMACRO_CPP, 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)

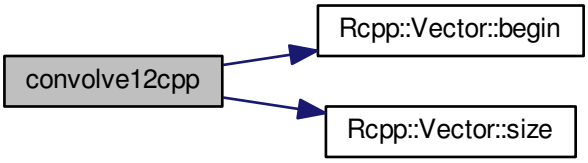
7.8.1 Function Documentation

7.8.1.1 RcppExport SEXP convolve12cpp (SEXP *a*, SEXP *b*)

Definition at line 7 of file convolve12_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), LOOPMACRO_CPP, 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 template<typename T > T convolve (const T & *a*, const T & *b*)

Definition at line 8 of file convolve13_cpp.cpp.

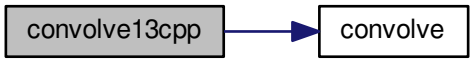
Referenced by convolve13cpp().

7.9.1.2 RcppExport SEXP convolve13cpp (SEXP *a*, SEXP *b*)

Definition at line 21 of file convolve13_cpp.cpp.

References convolve(), and LOOPMACRO_CPP.

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 RcppExport SEXP convolve14cpp (SEXP *a*, SEXP *b*)

Definition at line 8 of file convolve14_cpp.cpp.

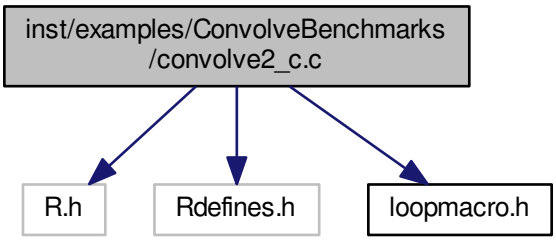
References LOOPMACRO_CPP, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

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 SEXP convolve2 (SEXP *a*, SEXP *b*)

Examples:

ConvolveBenchmarks/convolve2_c.c.

Definition at line 7 of file convolve2_c.c.

References LOOPMACRO_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 RcppExport SEXP convolve3cpp (SEXP *a*, SEXP *b*)

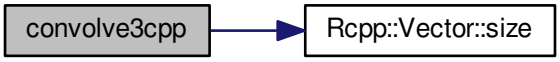
Examples:

ConvolveBenchmarks/convolve3_cpp.cpp.

Definition at line 7 of file convolve3_cpp.cpp.

References LOOPMACRO_CPP, and 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 RcppExport SEXP convolve4cpp (SEXP *a,* SEXP *b*)

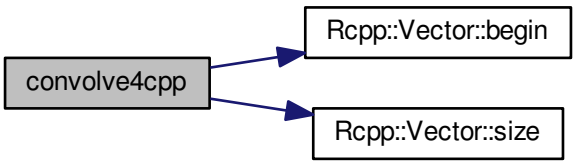
Examples:

ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 7 of file convolve4_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), LOOPMACRO_CPP, 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)

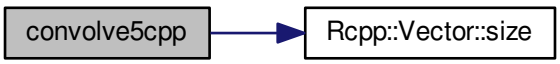
7.14.1 Function Documentation

7.14.1.1 RcppExport SEXP convolve5cpp (SEXP *a*, SEXP *b*)

Definition at line 9 of file convolve5_cpp.cpp.

References LOOPMACRO_CPP, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

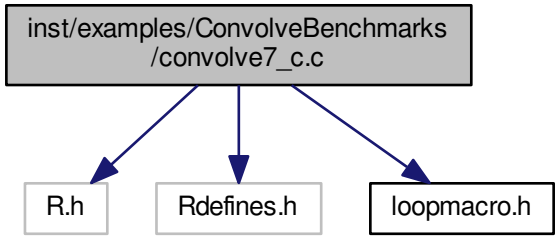
Here is the call graph for this function:



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 SEXP convolve7 (SEXP *a*, SEXP *b*)

Examples:

ConvolveBenchmarks/convolve7_c.c.

Definition at line 8 of file convolve7_c.c.

References LOOPMACRO_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)

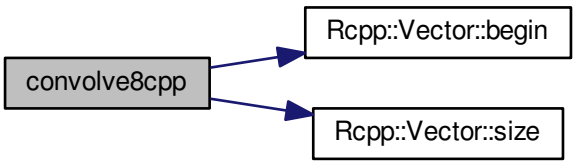
7.16.1 Function Documentation

7.16.1.1 RcppExport SEXP convolve8cpp (SEXP *a*, SEXP *b*)

Definition at line 24 of file convolve8_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), LOOPMACRO_CPP, 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)

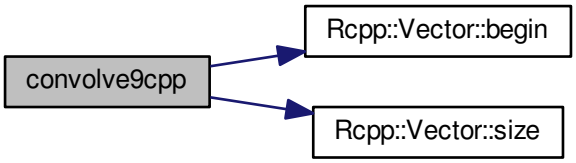
7.17.1 Function Documentation

7.17.1.1 RcppExport SEXP convolve9cpp (SEXP *a*, SEXP *b*)

Definition at line 42 of file convolve9_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), LOOPMACRO_CPP, 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 #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.

Referenced by convolve2(), and convolve7().

7.19.1.2 #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.

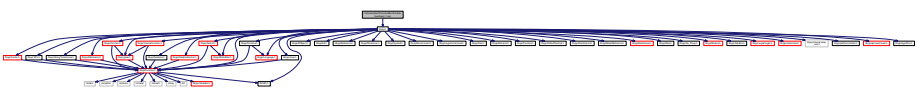
Referenced by convolve10cpp(), convolve11cpp(), convolve12cpp(), convolve13cpp(), convolve14cpp(), convolve3cpp(), convolve4cpp(), convolve5cpp(), convolve8cpp(), and convolve9cpp().

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 SEXP overhead_cpp (SEXP *a*, SEXP *b*)

Definition at line 8 of file overhead_1.cpp.

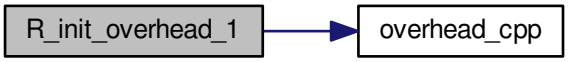
Referenced by R_init_overhead_1().

7.21.1.2 void R_init_overhead_1 (DllInfo * *info*)

Definition at line 12 of file overhead_1.cpp.

References overhead_cpp().

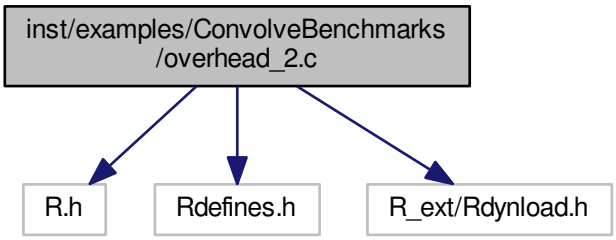
Here is the call graph for this function:



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 SEXP overhead_c (SEXP *a*, SEXP *b*)

Definition at line 8 of file overhead_2.c.

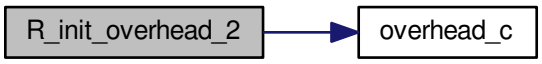
Referenced by R_init_overhead_2().

7.22.1.2 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)

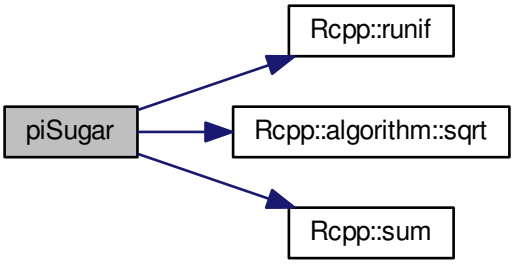
7.34.1 Function Documentation

7.34.1.1 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 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

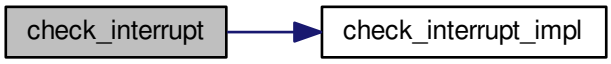
True, if a user interrupt has been detected.

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 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	*dummy*	Dummy argument.

Definition at line 50 of file piWithInterrupts.cpp.

Referenced by check_interrupt().

7.37.1.3 RcppExport SEXP PiLeibniz (SEXP *n*, SEXP *frequency*)

Compute pi using the Leibniz formula (a very inefficient approach).

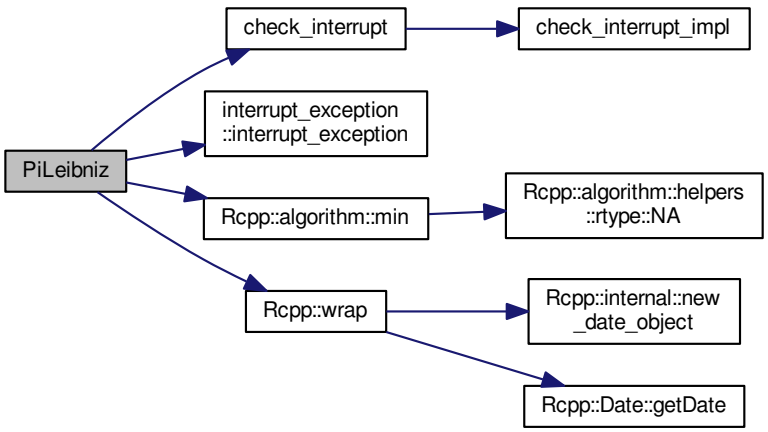
Parameters

in	*n*	Number of summands
in	*frequency*	Check for interrupts after every `frequency` loop cycles.

Definition at line 75 of file piWithInterrupts.cpp.

References BEGIN_RCPP, check_interrupt(), END_RCPP, interrupt_exception::interrupt_exception(), 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/timeRNGs.R File Reference

7.42 inst/examples/RcppInline/external_pointer.r File Reference

7.43 inst/examples/RcppInline/RcppInlineExample.r File Reference

7.44 inst/examples/RcppInline/RcppInlineWithLibsExamples.r File Reference

7.45 inst/examples/RcppInline/RcppSimpleExample.r File Reference

7.46 inst/examples/RcppInline/RObject.r File Reference

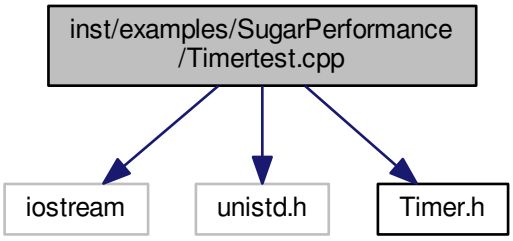
7.47 inst/examples/RcppInline/UncaughtExceptions.r File Reference

7.48 inst/examples/SugarPerformance/sugarBenchmarks.R File Reference

7.49 inst/examples/SugarPerformance/Timertest.cpp File Reference

```
#include <iostream>
#include <unistd.h>
#include "Timer.h"
```

Include dependency graph for Timertest.cpp:



Functions

- int main ()

7.49.1 Function Documentation

7.49.1.1 int main ( )

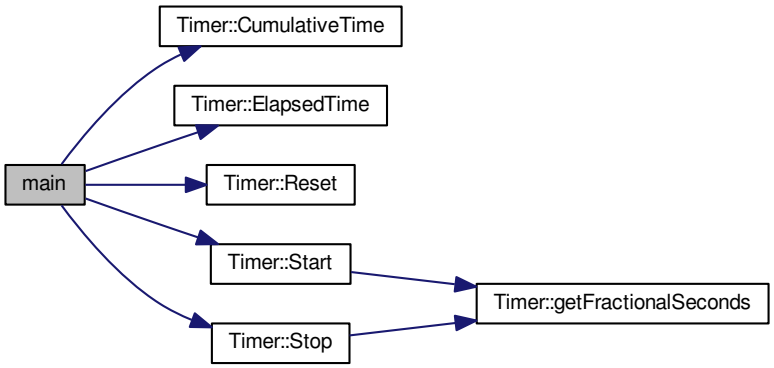
Examples:

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:



7.50 inst/include/doxygen/Examples.h File Reference

7.51 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/Date.h>
#include <Rcpp/DateVector.h>
#include <Rcpp/Datetime.h>
#include <Rcpp/DatetimeVector.h>
#include <Rcpp/Na_Proxy.h>
#include <Rcpp/Module.h>
#include <Rcpp/InternalFunction.h>
#include <Rcpp/Nullable.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>
```

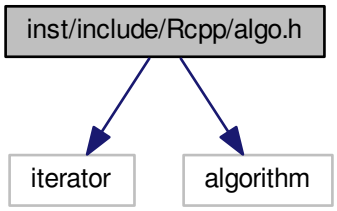
Include dependency graph for Rcpp.h:

This graph shows which files directly or indirectly include this file:

7.52 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

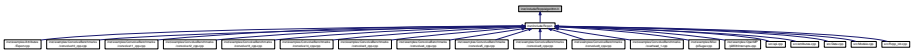
  Rcpp API.

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)

7.53 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 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)

7.53.1 Macro Definition Documentation

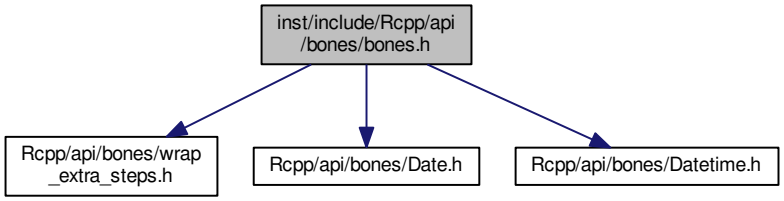
7.53.1.1 #define RCPP_CONSTEXPR const

Definition at line 4 of file algorithm.h.

7.54 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:

7.55 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

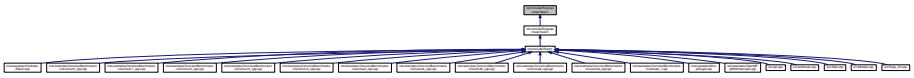
  internal implementation details

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)

7.56 inst/include/Rcpp/api/meat/Date.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

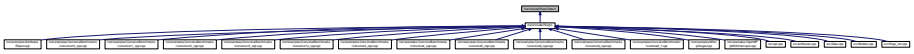
- Rcpp
  Rcpp API.

Functions

- template<>
  SEXP Rcpp::wrap (const Date &date)

7.57 inst/include/Rcpp/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)
- SEXP Rcpp::internal::getPosixClasses ()
- SEXP Rcpp::internal::new_posixt_object (double d)
- SEXP Rcpp::internal::new_date_object (double d)

7.58 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

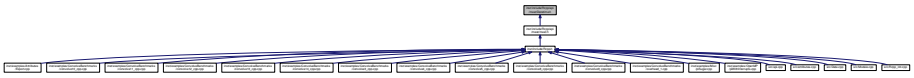
  internal implementation details

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)

7.59 inst/include/Rcpp/api/meat/Datetime.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

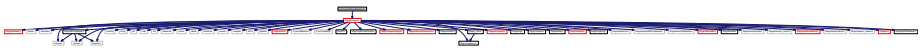
  Rcpp API.

Functions

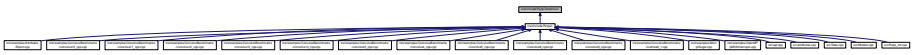
- template<>
  SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)
- template<>
  SEXP Rcpp::wrap< Datetime > (const Datetime &date)

7.60 inst/include/Rcpp/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)
- 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)

7.61 inst/include/Rcpp/api/bones/wrap_extra_steps.h File Reference

This graph shows which files directly or indirectly include this file:

Namespaces

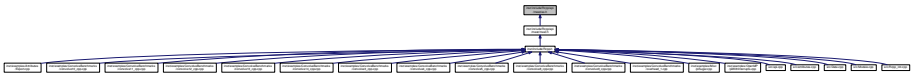
- Rcpp
  Rcpp API.

Functions

- template<typename T >
  SEXP Rcpp::wrap_extra_steps (SEXP x)

7.62 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

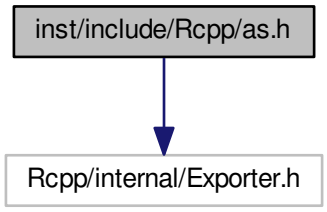
Functions

- void * Rcpp::internal::as_module_object_internal (SEXP obj)

7.63 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

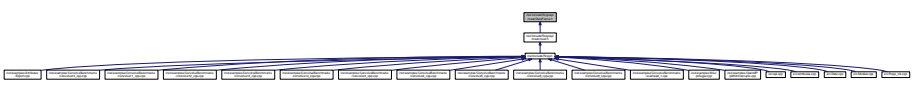
  internal implementation details

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)

7.64 inst/include/Rcpp/api/meat/DataFrame.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

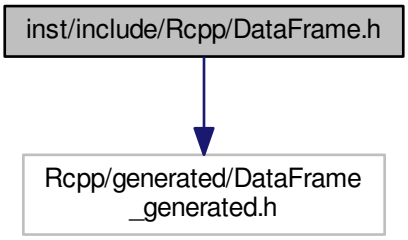
- Rcpp

  Rcpp API.

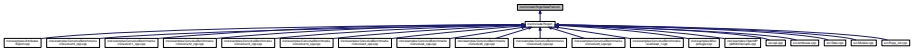
7.65 inst/include/Rcpp/DataFrame.h File Reference

```
#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

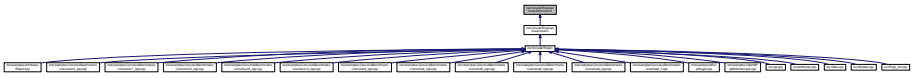
- typedef DataFrame_Impl< PreserveStorage > Rcpp::DataFrame

Functions

- SEXP Rcpp::internal::empty_data_frame ()

7.66 inst/include/Rcpp/api/meat/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:

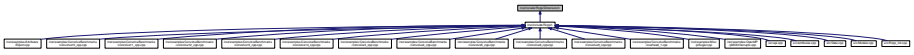


Namespaces

- Rcpp
  Rcpp API.

7.67 inst/include/Rcpp/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

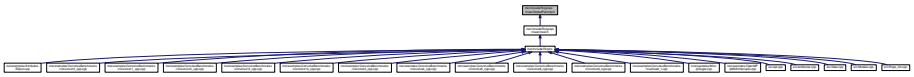
- class Rcpp::Dimension

Namespaces

- Rcpp
  Rcpp API.

7.68 inst/include/Rcpp/api/meat/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:



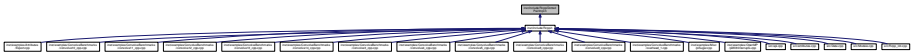
Namespaces

- Rcpp

  Rcpp API.

7.69 inst/include/Rcpp/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DottedPairImpl< CLASS >

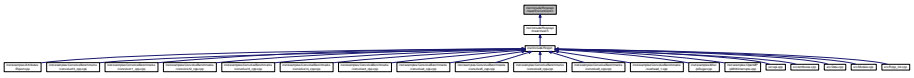
Namespaces

- Rcpp

  Rcpp API.

7.70 inst/include/Rcpp/api/meat/Environment.h File Reference

This graph shows which files directly or indirectly include this file:



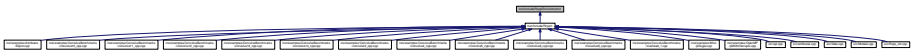
Namespaces

- Rcpp

  Rcpp API.

7.71 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

- SEXP R_NewHashedEnv (SEXP, SEXP)
- 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
- 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::bindingIsLocked (const std::string &name) const
- bool Rcpp::bindingIsActive (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)
- void Rcpp::update (SEXP)
- Environment Rcpp::new_env (int size=29)
- Environment Rcpp::new_env (SEXP parent, int size=29)

Variables

- public Rcpp::BindingPolicy< Environment_Impl< StoragePolicy > >

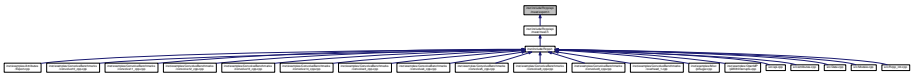
7.71.1 Function Documentation

7.71.1.1 SEXP R_NewHashedEnv (SEXP , SEXP)

Referenced by Rcpp::new_env().

7.72 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

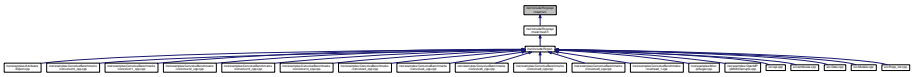
  traits used to dispatch wrap

Functions

- template<typename InputIterator , typename value_type >
  void Rcpp::internal::export_range__dispatch (SEXP x, InputIterator first,::Rcpp::traits::r_type_generic_tag)

7.73 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)

7.74 inst/include/Rcpp/is.h File Reference

This graph shows which files directly or indirectly include this file:

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

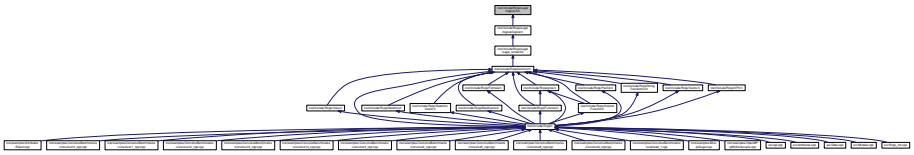
  internal implementation details

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)

7.75 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

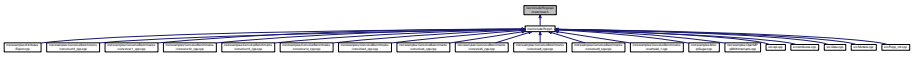
- template<bool NA, typename T >
  bool Rcpp::is_true (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool Rcpp::is_false (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool Rcpp::is_na (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)

7.76 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/module/Module.h>
```

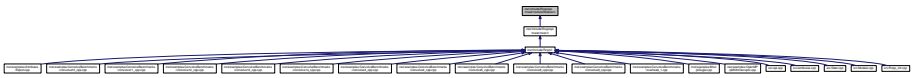
Include dependency graph for meat.h:

This graph shows which files directly or indirectly include this file:



7.77 inst/include/Rcpp/api/meat/module/Module.h File Reference

This graph shows which files directly or indirectly include this file:



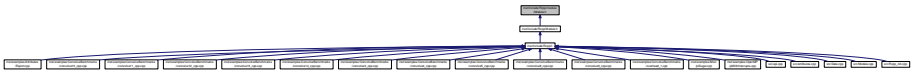
Namespaces

- Rcpp

  Rcpp API.

7.78 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

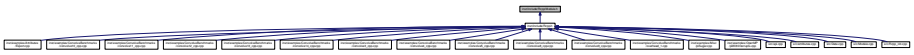
  Rcpp API.

7.79 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::CppInheritedMethod< 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::CppProperty< Class >
- class Rcpp::CppInheritedProperty< Class, Parent >
- class Rcpp::CppFinalizer< 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

7.79.1 Macro Definition Documentation

7.79.1.1 #define GET_MODULE_SYM (moduleSym == NULL ? moduleSym = Rf_install("Module") : moduleSym)

Definition at line 452 of file Module.h.

7.79.1.2 #define LOAD_RCPP_MODULE(*NAME*)

Value:

```
Shield<SEXP> __load_module_call__( Rf_lang2( GET_MODULE_SYM, _rcpp_module_boot_##NAME() );
      \
    Rcpp_eval(__load_module_call__), R_GlobalEnv );
```

Definition at line 455 of file Module.h.

7.79.1.3 #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.

7.79.1.4 #define RCPP_MODULE_BOOT(*name*) _rcpp_module_boot_##name

Definition at line 426 of file Module.h.

7.79.1.5 #define VARIABLE_IS_NOT_USED

Definition at line 444 of file Module.h.

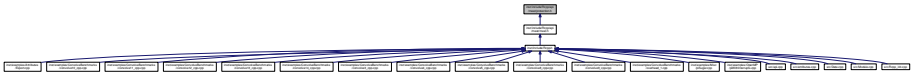
7.79.2 Variable Documentation

7.79.2.1 VARIABLE_IS_NOT_USED SEXP moduleSym = NULL [static]

Definition at line 448 of file Module.h.

7.80 inst/include/Rcpp/api/meat/protection.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

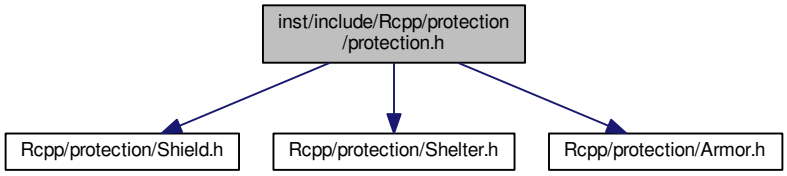
- Rcpp

  Rcpp API.

7.81 inst/include/Rcpp/protection/protection.h File Reference

```
#include <Rcpp/protection/Shield.h>
#include <Rcpp/protection/Shelter.h>
#include <Rcpp/protection/Armor.h>
```

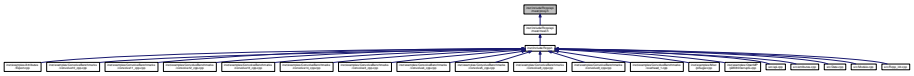
Include dependency graph for protection.h:



This graph shows which files directly or indirectly include this file:

7.82 inst/include/Rcpp/api/meat/proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

7.83 inst/include/Rcpp/proxy/proxy.h File Reference

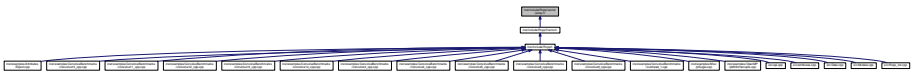
```
#include <Rcpp/proxy/GenericProxy.h>
#include <Rcpp/proxy/NamesProxy.h>
#include <Rcpp/proxy/RObjectMethods.h>
#include <Rcpp/proxy/AttributeProxy.h>
#include <Rcpp/proxy/TagProxy.h>
#include <Rcpp/proxy/ProtectedProxy.h>
#include <Rcpp/proxy/SlotProxy.h>
#include <Rcpp/proxy/Binding.h>
#include <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:

7.84 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 >
- class Rcpp::internal::string_name_proxy< RTYPE >
- class Rcpp::internal::generic_name_proxy< RTYPE >
- struct Rcpp::traits::r_vector_name_proxy< RTYPE >
- struct Rcpp::traits::r_vector_name_proxy< STRSXP >
- struct Rcpp::traits::r_vector_name_proxy< VECSXP >
- struct Rcpp::traits::r_vector_name_proxy< EXPRSXP >
- struct Rcpp::traits::r_vector_proxy< RTYPE >
- struct Rcpp::traits::r_vector_proxy< STRSXP >
- struct Rcpp::traits::r_vector_proxy< EXPRSXP >
- struct Rcpp::traits::r_vector_proxy< VECSXP >
- struct Rcpp::traits::r_vector_const_proxy< RTYPE >
- struct Rcpp::traits::r_vector_const_proxy< STRSXP >
- struct Rcpp::traits::r_vector_const_proxy< VECSXP >
- struct Rcpp::traits::r_vector_const_proxy< EXPRSXP >
- struct Rcpp::traits::r_vector_iterator< RTYPE >
- struct Rcpp::traits::r_vector_const_iterator< RTYPE >
- struct Rcpp::traits::proxy_based_iterator< RTYPE >
- struct Rcpp::traits::r_vector_iterator< VECSXP >
- struct Rcpp::traits::r_vector_iterator< EXPRSXP >
- struct Rcpp::traits::r_vector_iterator< STRSXP >
- struct Rcpp::traits::proxy_based_const_iterator< RTYPE >
- struct Rcpp::traits::r_vector_const_iterator< VECSXP >
- struct Rcpp::traits::r_vector_const_iterator< EXPRSXP >
- struct Rcpp::traits::r_vector_const_iterator< STRSXP >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

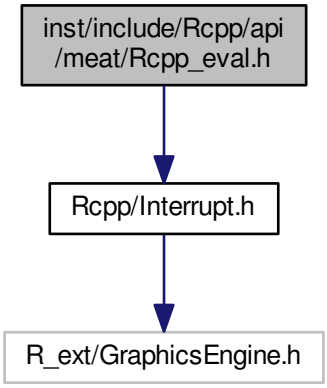
  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

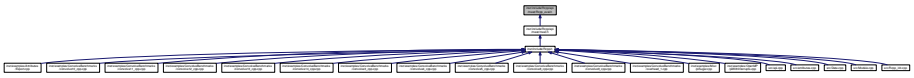
7.85 inst/include/Rcpp/api/meat/Rcpp_eval.h File Reference

```
#include <Rcpp/Interrupt.h>
```

Include dependency graph for Rcpp_eval.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

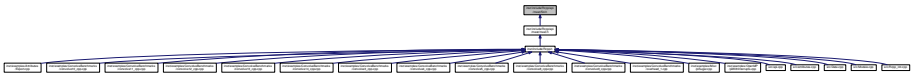
  Rcpp API.

Functions

- SEXP Rcpp::Rcpp_eval (SEXP expr, SEXP env)

7.86 inst/include/Rcpp/api/meat/S4.h File Reference

This graph shows which files directly or indirectly include this file:



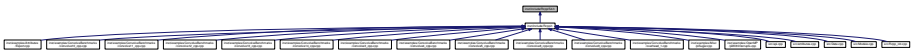
Namespaces

- Rcpp

  Rcpp API.

7.87 inst/include/Rcpp/S4.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

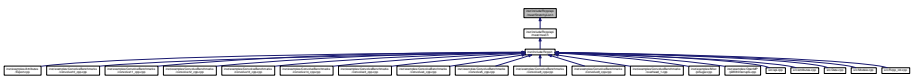
- typedef S4_Impl< PreserveStorage > Rcpp::S4

Functions

- Rcpp::RCPP_API_CLASS (S4_Impl)

7.88 inst/include/Rcpp/api/meat/StretchyList.h File Reference

This graph shows which files directly or indirectly include this file:



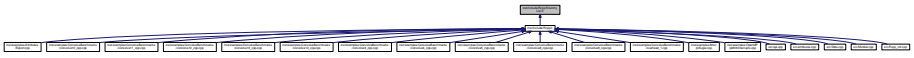
Namespaces

- Rcpp

  Rcpp API.

7.89 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

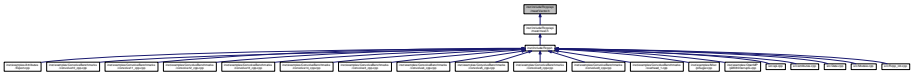
- Rcpp::RCPP_API_CLASS (StretchyList_Impl)
- Rcpp::StretchyList_Impl ()
- Rcpp::StretchyList_Impl (SEXP x)
- void Rcpp::update (SEXP)
- Rcpp::operator SEXP () const
- template<typename T >
  StretchyList_Impl & Rcpp::push_back (const T &obj)
- template<typename T >
  StretchyList_Impl & Rcpp::push_front (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 > >

7.90 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

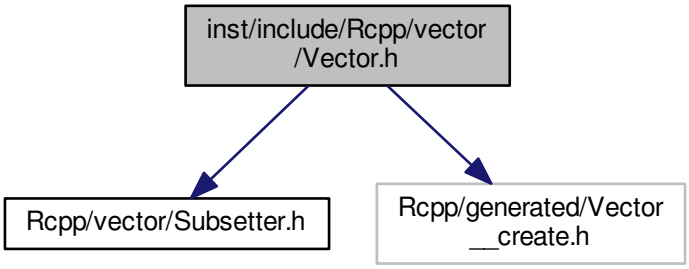
Functions

- template<typename T >
  SEXP Rcpp::internal::wrap_range_sugar_expression (const T &object, Rcpp::traits::true_type)

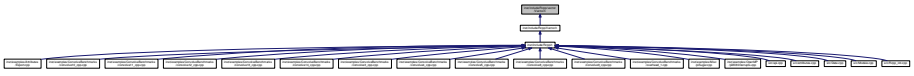
7.91 inst/include/Rcpp/vector/Vector.h File Reference

```
#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

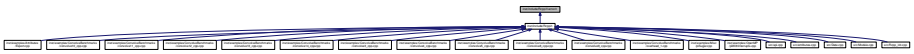
- template<int RTYPE, template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< RTYPE, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< STRSXP, StoragePolicy > &rhs)

7.92 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:



Classes

- struct Rcpp::traits::Extractor< RTYPE, NA, VECTOR >
- class Rcpp::MatrixRow< RTYPE >
- class Rcpp::ConstMatrixRow< RTYPE >
- class Rcpp::MatrixColumn< RTYPE >
- class Rcpp::ConstMatrixColumn< RTYPE >
- class Rcpp::SubMatrix< RTYPE >
- class Rcpp::SingleLogicalResult< NA, T >

Namespaces

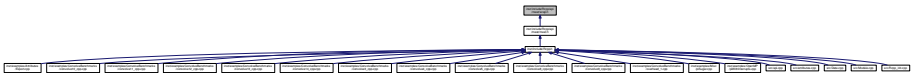
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.93 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)

7.94 inst/include/Rcpp/barrier.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- SEXP get_string_elt (SEXP, int)
- const char * char_get_string_elt (SEXP, int)
- void set_string_elt (SEXP, int, SEXP)
- void char_set_string_elt (SEXP, int, const char *)
- SEXP * get_string_ptr (SEXP)
- SEXP get_vector_elt (SEXP, int)
- void set_vector_elt (SEXP, int, SEXP)
- SEXP * get_vector_ptr (SEXP)
- const char * char_nocheck (SEXP)
- void * dataptr (SEXP)

7.94.1 Function Documentation

7.94.1.1 const char* char_get_string_elt (SEXP , int) [inline]

Definition at line 158 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.2 const char* char_nocheck (SEXP) [inline]

Definition at line 200 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::String::setBuffer().

7.94.1.3 void char_set_string_elt (SEXP , int , const char *) [inline]

Definition at line 170 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.4 void* dataptr (SEXP) [inline]

Definition at line 206 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.5 SEXP get_string_elt (SEXP , int) [inline]

Definition at line 152 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.6 SEXP* get_string_ptr (SEXP) [inline]

Definition at line 176 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::sugar::get_const_begin(), and registerFunctions().

7.94.1.7 SEXP get_vector_elt (SEXP , int) [inline]

Definition at line 182 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.8 SEXP* get_vector_ptr (SEXP) [inline]

Definition at line 194 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.9 void set_string_elt (SEXP , int , SEXP) [inline]

Definition at line 164 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.94.1.10 void set_vector_elt (SEXP , int , SEXP) [inline]

Definition at line 188 of file routines.h.

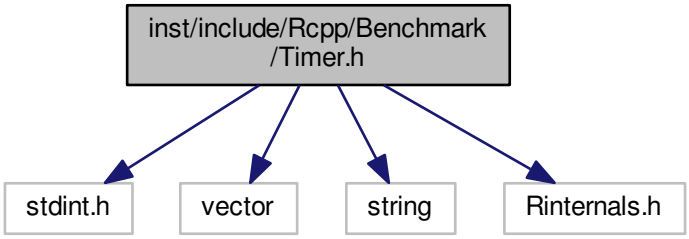
References GET_CALLABLE.

Referenced by registerFunctions().

7.95 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

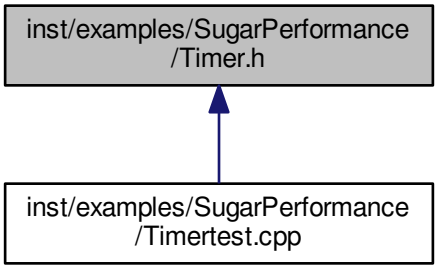
7.95.1 Macro Definition Documentation

7.95.1.1 #define R_NO_REMAP

Definition at line 29 of file Timer.h.

7.96 inst/examples/SugarPerformance/Timer.h File Reference

This graph shows which files directly or indirectly include this file:



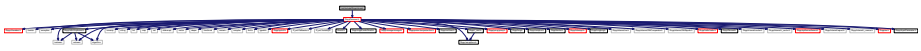
Classes

- class Timer

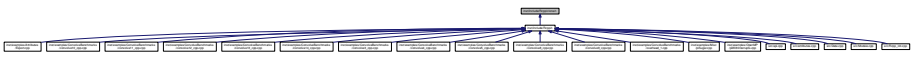
7.97 inst/include/Rcpp/clone.h File Reference

```
#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)

7.98 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)

7.98.1 Macro Definition Documentation

7.98.1.1 #define dplyr_tools_complex_H

Definition at line 74 of file complex.h.

7.98.2 Function Documentation

7.98.2.1 Rcomplex operator∗ (const Rcomplex & *lhs,* const Rcomplex & *rhs*) [inline]

Definition at line 25 of file complex.h.

7.98.2.2 Rcomplex operator+ (const Rcomplex & *lhs,* const Rcomplex & *rhs*) [inline]

Definition at line 32 of file complex.h.

7.98.2.3 Rcomplex operator- (const Rcomplex & *lhs,* const Rcomplex & *rhs*) [inline]

Definition at line 39 of file complex.h.

7.98.2.4 Rcomplex operator/ (const Rcomplex & *a,* const Rcomplex & *b*) [inline]

Definition at line 46 of file complex.h.

7.98.2.5 std::ostream& operator<< (std::ostream & *os,* const Rcomplex & *cplx*) [inline]

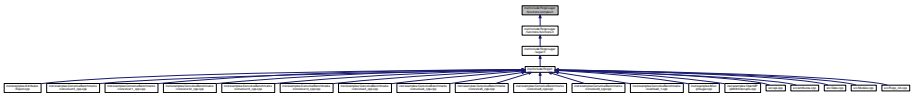
Definition at line 76 of file complex.h.

7.98.2.6 bool operator== (const Rcomplex & *a,* const Rcomplex & *b*) [inline]

Definition at line 68 of file complex.h.

7.99 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_HYPOT ::Rf_pythag
- #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)
- 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)

7.99.1 Macro Definition Documentation

7.99.1.1 #define RCPP_HYPOT ::Rf_pythag

Definition at line 28 of file complex.h.

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

7.99.1.2 #define RCPP_SUGAR_COMPLEX(*__NAME__, __OUT__*)

Value:

```
template <bool NA, typename T>                                          \
        inline sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*)(Rcomplex) >        \
        __NAME__(                                                               \
                const VectorBase<CPLXSXP,NA,T>& t                                       \
                ){                                                              \
                return sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*)(Rcomplex) >(   \
                        internal::complex__##__NAME__, t                                \
                ) ;                                                             \
        }
```

Definition at line 238 of file complex.h.

7.100 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(0,12,7)
- #define RCPP_DEV_VERSION RcppDevVersion(0,12,7,0)

7.100.1 Macro Definition Documentation

7.100.1.1 #define RCPP_DEV_VERSION RcppDevVersion(0,12,7,0)

Definition at line 33 of file config.h.

7.100.1.2 #define Rcpp_Version(*v, p, s*) (((v) ∗ 65536) + ((p) ∗ 256) + (s))

Definition at line 25 of file config.h.

7.100.1.3 #define RCPP_VERSION Rcpp_Version(0,12,7)

Definition at line 30 of file config.h.

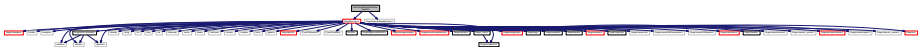
7.100.1.4 #define RcppDevVersion(*maj, min, rev, dev*) (((maj)∗1000000) + ((min)∗10000) + ((rev)∗100) + (dev))

Definition at line 27 of file config.h.

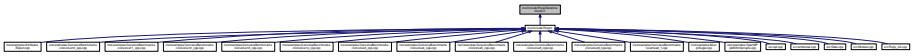
7.101 inst/include/Rcpp/DatetimeVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
```

Include dependency graph for DatetimeVector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DatetimeVector

Namespaces

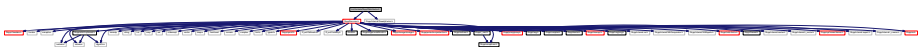
- Rcpp

  Rcpp API.

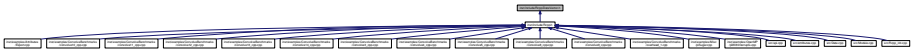
7.102 inst/include/Rcpp/DateVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
```

Include dependency graph for DateVector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DateVector

Namespaces

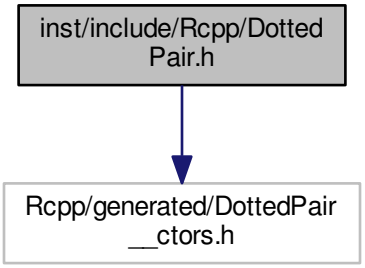
- Rcpp

  Rcpp API.

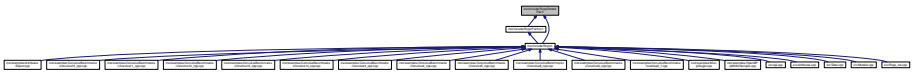
7.103 inst/include/Rcpp/DottedPair.h File Reference

```
#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

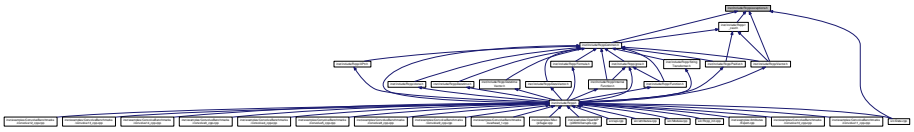
- Rcpp::RCPP_API_CLASS (DottedPair_Impl)

Variables

- public Rcpp::DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< DottedPair_Impl< StoragePolicy > >

7.104 inst/include/Rcpp/exceptions.h File Reference

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

Namespaces

- Rcpp

  Rcpp API.

Macros

- #define GET_STACKTRACE() stack_trace(__FILE__, __LINE__)
- #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)
- 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)
- 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 Rcpp::warning (const std::string &message)
- 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)
- void NORET Rcpp::stop (const std::string &message)
- 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)
- void forward_exception_to_r (const std::exception &ex)

7.104.1 Macro Definition Documentation

7.104.1.1 #define DEMANGLE(__TYPE__) demangle(typeid(__TYPE__).name()).c_str()

Definition at line 209 of file exceptions.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::String::append_wide_string(), Rcpp::internal::as(), Rcpp::finalizer_wrapper(), Rcpp::internal::string_element_converter< RTYPE >::get(), class_< Class >::get_instance(), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse(), Rcpp::Vector< INTSXP >::import_sugar_expression(), Rcpp::internal::range_wrap_dispatch___impl__pair(), Rcpp::Vector< INTSXP >::replace_element__dispatch__isArgument(), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::Sapply(), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::Sapply(), Rcpp::Vector< INTSXP >::Vector(), and Rcpp::internal::wrap_range_sugar_expression().

7.104.1.2 #define GET_STACKTRACE( ) stack_trace(__FILE__, __LINE__)

Definition at line 25 of file exceptions.h.

7.104.1.3 #define RCPP_EXCEPTION_CLASS(__CLASS__, __WHAT__)

Value:

```
class __CLASS__ : public std::exception{                                    \
    public:                                                                 \
        __CLASS__( const std::string& message ) throw() : message( __WHAT__ ){} ;  \
        virtual ~__CLASS__() throw(){} ;                                    \
        virtual const char* what() const throw() { return message.c_str() ; }  \
    private:                                                                \
        std::string message ;                                               \
    } ;
```

Definition at line 83 of file exceptions.h.

7.104.1.4 #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 93 of file exceptions.h.

7.104.2 Function Documentation

7.104.2.1 std::string demangle (const std::string & *name*) [inline]

Definition at line 134 of file routines.h.

References GET_CALLABLE.

Referenced by exception_to_r_condition(), exception_to_try_error(), Rcpp::get_return_type_dispatch(), and register↩ Functions().

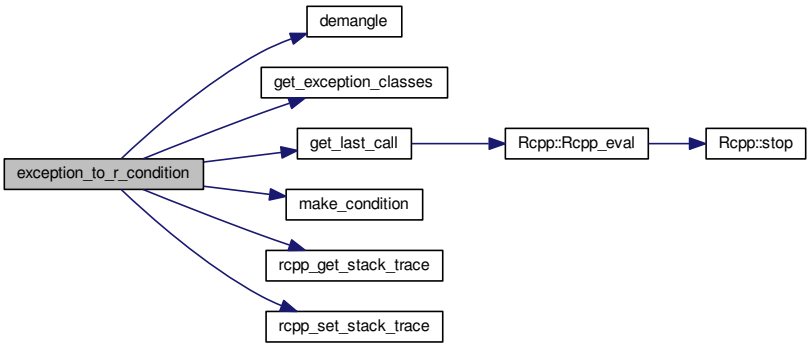
7.104.2.2 SEXP exception_to_r_condition (const std::exception & *ex*) [inline]

Definition at line 173 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 forward_exception_to_r().

Here is the call graph for this function:

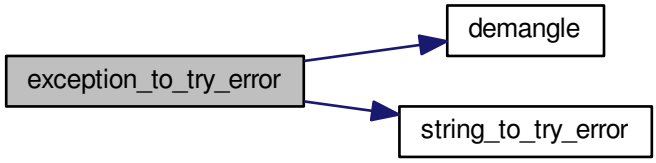


7.104.2.3 SEXP exception_to_try_error (const std::exception & *ex*) [inline]

Definition at line 204 of file exceptions.h.

References demangle(), and string_to_try_error().

Here is the call graph for this function:

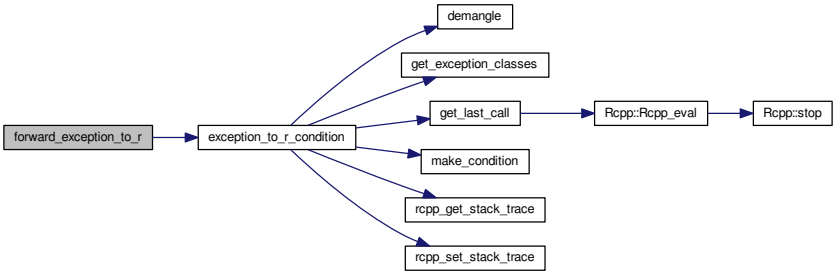


7.104.2.4 void forward_exception_to_r (const std::exception & *ex*) [inline]

Definition at line 322 of file exceptions.h.

References exception_to_r_condition().

Here is the call graph for this function:



7.104.2.5 SEXP get_exception_classes (const std::string & *ex_class*) [inline]

Definition at line 138 of file exceptions.h.

Referenced by exception_to_r_condition().

7.104.2.6 SEXP get_last_call ( ) [inline]

Definition at line 129 of file exceptions.h.

References Rcpp::Rcpp_eval().

Referenced by exception_to_r_condition().

Here is the call graph for this function:

7.104.2.7 SEXP make_condition (const std::string & *ex_msg*, SEXP *call*, SEXP *cppstack*, SEXP *classes*) [inline]

Definition at line 152 of file exceptions.h.

Referenced by exception_to_r_condition().

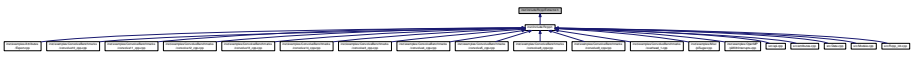
7.104.2.8 SEXP string_to_try_error (const std::string & *str*) [inline]

Definition at line 185 of file exceptions.h.

Referenced by exception_to_try_error().

7.105 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

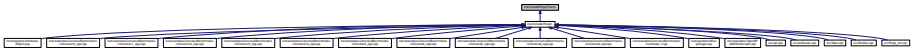
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.106 inst/include/Rcpp/Fast.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Fast< VECTOR >

Namespaces

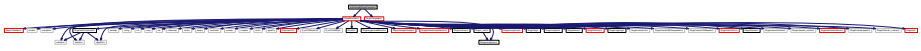
- Rcpp

  Rcpp API.

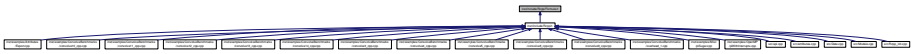
7.107 inst/include/Rcpp/Formula.h File Reference

```
#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

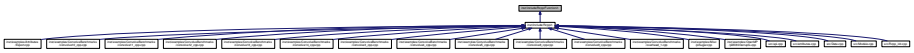
- public Rcpp::DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< Formula_Impl< StoragePolicy > >

7.108 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

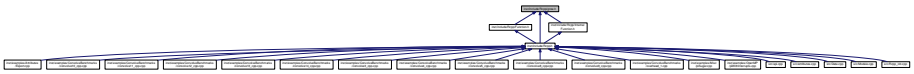
- Rcpp::RCPP_API_CLASS (Function_Impl)

7.109 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

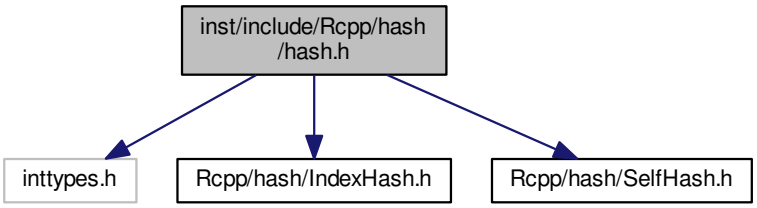
Functions

- 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)

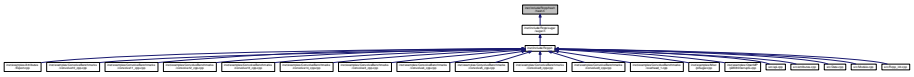
7.110 inst/include/Rcpp/hash/hash.h File Reference

```
#include <inttypes.h>
#include <Rcpp/hash/IndexHash.h>
#include <Rcpp/hash/SelfHash.h>
```

Include dependency graph for hash.h:

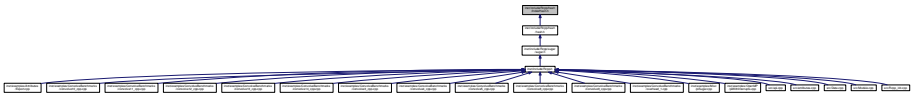


This graph shows which files directly or indirectly include this file:



7.111 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 * ((unsigned int)(X)) >> (32 - k))

7.111.1 Macro Definition Documentation

7.111.1.1 #define RCPP_HASH(*X*) (3141592653U * ((unsigned int)(X)) >> (32 - k))

Definition at line 46 of file IndexHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::get_addr(), and Rcpp::sugar::IndexHash< RTYPE >::get_addr().

7.111.1.2 #define RCPP_PROFILE_RECORD(*name*)

Definition at line 38 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::lookup__impl().

7.111.1.3 #define RCPP_PROFILE_TIC

Definition at line 36 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::lookup__impl().

7.111.1.4 #define RCPP_PROFILE_TOC

Definition at line 37 of file IndexHash.h.

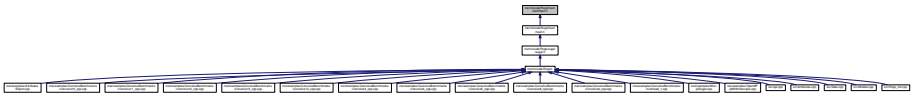
Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::lookup__impl().

7.111.1.5 #define RCPP_USE_CACHE_HASH

Definition at line 40 of file IndexHash.h.

7.112 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

7.113 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

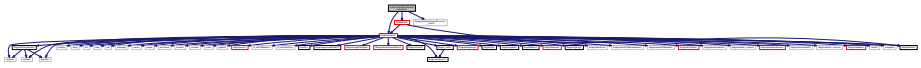
  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

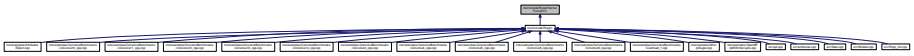
7.114 inst/include/Rcpp/InternalFunction.h File Reference

```
#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

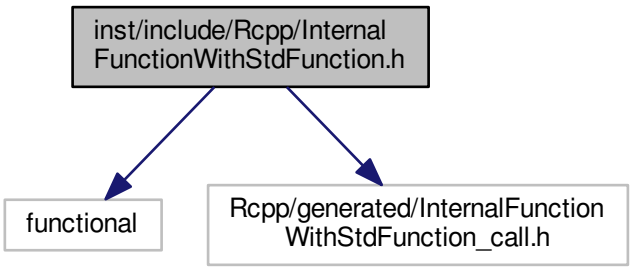
Functions

- Rcpp::RCPP_API_CLASS (InternalFunction_Impl)

7.115 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... >

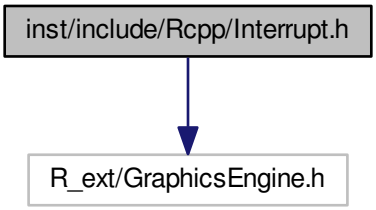
Namespaces

- Rcpp
  Rcpp API.
- Rcpp::InternalFunctionWithStdFunction

7.116 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

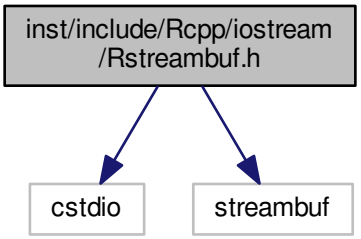
Functions

- SEXP Rcpp::internal::interruptedError ()
- void Rcpp::checkUserInterrupt ()

7.117 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
- static Rostream< false > Rcpp::Rcerr

7.118 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

Functions

- SEXP Rcpp::Rcpp_lcons (SEXP car, SEXP cdr)
- SEXP Rcpp::Rcpp_list2 (SEXP x0, SEXP x1)
- SEXP Rcpp::Rcpp_lang2 (SEXP x0, SEXP x1)
- SEXP Rcpp::Rcpp_list3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp::Rcpp_lang3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp::Rcpp_list4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP Rcpp::Rcpp_lang4 (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_lang5 (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_lang6 (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)

7.118.1 Macro Definition Documentation

7.118.1.1 #define Rcpp_lang1 Rf_lang1

Definition at line 26 of file lang.h.

7.118.1.2 #define Rcpp_list1 Rf_list1

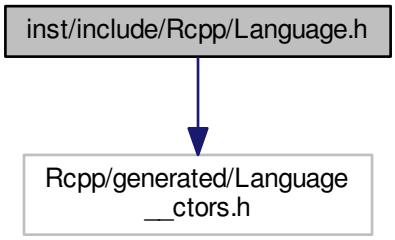
Definition at line 25 of file lang.h.

Referenced by Rcpp::Rcpp_lang2(), and Rcpp::Rcpp_list2().

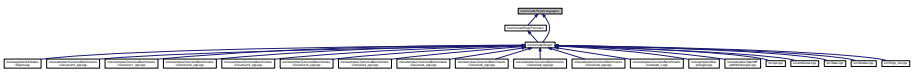
7.119 inst/include/Rcpp/Language.h File Reference

```
#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 > >

7.120 inst/include/Rcpp/longlong.h File Reference

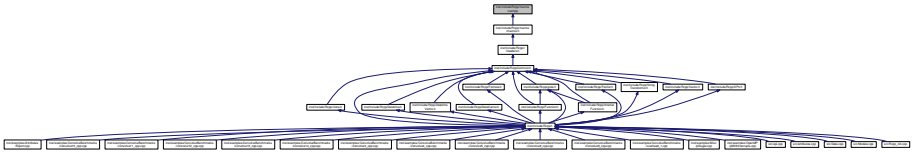
This graph shows which files directly or indirectly include this file:

7.121 inst/include/Rcpp/traits/longlong.h File Reference

This graph shows which files directly or indirectly include this file:

7.122 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

7.122.1 Macro Definition Documentation

7.122.1.1 #define RCPP_PP_CAT(*a, b*) RCPP_PP_CAT_OO((a, b))

Definition at line 22 of file cat.hpp.

7.122.1.2 #define RCPP_PP_CAT_I(*a, b*) RCPP_PP_CAT_II(a ## b)

Definition at line 29 of file cat.hpp.

7.122.1.3 #define RCPP_PP_CAT_II(*res*) res

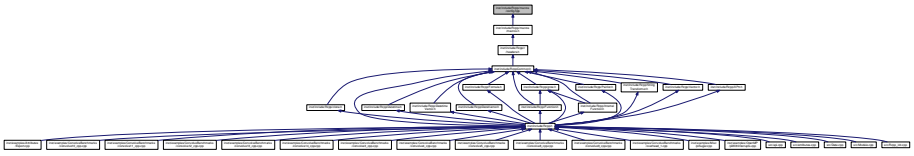
Definition at line 30 of file cat.hpp.

7.122.1.4 #define RCPP_PP_CAT_OO(*par*) RCPP_PP_CAT_I ## par

Definition at line 23 of file cat.hpp.

7.123 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

7.123.1 Macro Definition Documentation

7.123.1.1 #define RCPP_PP_CONFIG_BCC( ) 0x0010

Definition at line 22 of file config.hpp.

7.123.1.2 #define RCPP_PP_CONFIG_DMC( ) 0x0040

Definition at line 24 of file config.hpp.

7.123.1.3 #define RCPP_PP_CONFIG_EDG( ) 0x0020

Definition at line 23 of file config.hpp.

7.123.1.4 #define RCPP_PP_CONFIG_ERRORS 1

Definition at line 66 of file config.hpp.

7.123.1.5 #define RCPP_PP_CONFIG_EXTENDED_LINE_INFO 0

Definition at line 57 of file config.hpp.

7.123.1.6 #define RCPP_PP_CONFIG_FLAGS( ) (RCPP_PP_CONFIG_STRICT())

Definition at line 50 of file config.hpp.

7.123.1.7 #define RCPP_PP_CONFIG_IDEAL( ) 0x0002

Definition at line 18 of file config.hpp.

7.123.1.8 #define RCPP_PP_CONFIG_MSVC( ) 0x0004

Definition at line 20 of file config.hpp.

7.123.1.9 #define RCPP_PP_CONFIG_MWCC( ) 0x0008

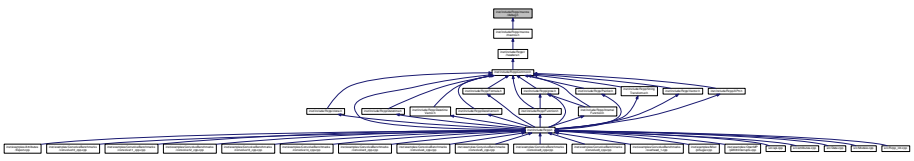
Definition at line 21 of file config.hpp.

7.123.1.10 #define RCPP_PP_CONFIG_STRICT( ) 0x0001

Definition at line 17 of file config.hpp.

7.124 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)

7.124.1 Macro Definition Documentation

7.124.1.1 #define RCPP_DEBUG(*MSG*)

Definition at line 43 of file debug.h.

Referenced by Rcpp::internal::string_element_converter< RTYPE >::get(), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse(), Rcpp::S4_CppConstructor< Class >::S4_CppConstructor(), and Rcpp::S4_field< Class >::S4_field().

7.124.1.2 #define RCPP_DEBUG_1(*fmt, MSG*)

Definition at line 44 of file debug.h.

Referenced by Rcpp::internal::as(), Rcpp::internal::string_element_converter< RTYPE >::get(), Rcpp::internal::generic_name_proxy< RTYPE >::operator T(), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::Sapply(), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::Sapply(), Rcpp::Vector< INTSXP >::Vector(), and Rcpp::internal::wrap_range_sugar_expression().

7.124.1.3 #define RCPP_DEBUG_2(*fmt, M1, M2*)

Definition at line 45 of file debug.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::Vector< INTSXP >::init(), Rcpp::Vector< INTSXP >::replace_element__dispatch__isArgument(), Rcpp::Vector< INTSXP >::Vector(), and Rcpp::XPtr< T, StoragePolicy, Finalizer >::XPtr().

7.124.1.4 #define RCPP_DEBUG_3(*fmt, M1, M2, M3*)

Definition at line 46 of file debug.h.

Referenced by Rcpp::finalizer_wrapper(), and Rcpp::internal::range_wrap_dispatch___impl__pair().

7.124.1.5 #define RCPP_DEBUG_4(*fmt, M1, M2, M3, M4*)

Definition at line 47 of file debug.h.

Referenced by Rcpp::MatrixRow< RTYPE >::get_parent_index(), Rcpp::ConstMatrixRow< RTYPE >::get_parent_↩ index(), and Rcpp::Vector< INTSXP >::import_sugar_expression().

7.124.1.6 #define RCPP_DEBUG_5(*fmt, M1, M2, M3, M4, M5*)

Definition at line 48 of file debug.h.

7.124.1.7 #define RCPP_DEBUG_LEVEL 0

Definition at line 27 of file debug.h.

7.124.1.8 #define RCPP_DEBUG_MODULE(*MSG*)

Definition at line 101 of file debug.h.

Referenced by class_< Class >::getMethods().

7.124.1.9 #define RCPP_DEBUG_MODULE_1(*fmt, MSG*)

Definition at line 102 of file debug.h.

Referenced by class_< Class >::AddMethod().

7.124.1.10 #define RCPP_DEBUG_MODULE_2(*fmt, M1, M2*)

Definition at line 103 of file debug.h.

Referenced by class_< Class >::get_instance().

7.124.1.11 #define RCPP_DEBUG_MODULE_3(*fmt, M1, M2, M3*)

Definition at line 104 of file debug.h.

7.124.1.12 #define RCPP_DEBUG_MODULE_4(*fmt, M1, M2, M3, M4*)

Definition at line 105 of file debug.h.

7.124.1.13 #define RCPP_DEBUG_MODULE_5(*fmt, M1, M2, M3, M4, M5*)

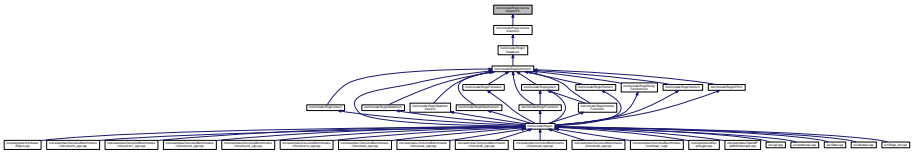
Definition at line 106 of file debug.h.

7.124.1.14 #define RCPP_DEBUG_MODULE_LEVEL RCPP_DEBUG_LEVEL

Definition at line 31 of file debug.h.

7.125 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)

7.125.1 Macro Definition Documentation

7.125.1.1 #define ___RCPP_HANDLE_CASE___(*___RTYPE___, ___FUN___, ___OBJECT___, ___RCPPTYPE___*)

Value:

```
case ___RTYPE___:                                                       \
    return ___FUN___(::Rcpp::___RCPPTYPE___<___RTYPE___>(___OBJECT___));
```

Definition at line 60 of file dispatch.h.

7.125.1.2 #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 65 of file dispatch.h.

7.125.1.3 #define RCPP_RETURN_MATRIX(_FUN_, _SEXP_) ___RCPP_RETURN___(_FUN_, _SEXP_, Matrix)

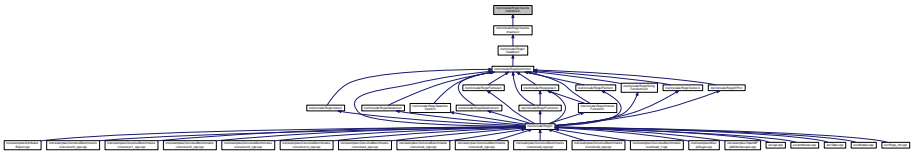
Definition at line 82 of file dispatch.h.

7.125.1.4 #define RCPP_RETURN_VECTOR(_FUN_, _SEXP_) ___RCPP_RETURN___(_FUN_, _SEXP_, Vector)

Definition at line 80 of file dispatch.h.

7.126 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__)

7.126.1 Macro Definition Documentation

7.126.1.1 #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 RObjectMethods< __CLASS__<StoragePolicy> >
```

Definition at line 49 of file interface.h.

7.126.1.2 #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.

7.126.1.3 #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.

Referenced by Rcpp::CppClass::CppClass(), Rcpp::CppObject::CppObject(), Rcpp::S4_CppConstructor< Class >::↩
S4_CppConstructor(), Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods(), and Rcpp::S4_↩
field< Class >::S4_field().

7.126.1.4 #define RCPP_GENERATE_CTOR_ASSIGN(__CLASS__)

Value:

```
typedef StoragePolicy<__CLASS__> Storage ;                                  \
typedef AttributeProxyPolicy<__CLASS__> AttributePolicy ;                   \
RCPP_CTOR_ASSIGN(__CLASS__)
```

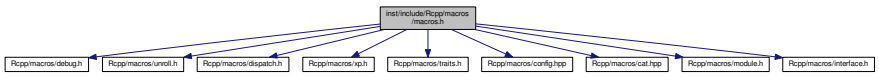
Definition at line 21 of file interface.h.

Referenced by Rcpp::RCPP_API_CLASS().

7.127 inst/include/Rcpp/macros/macros.h File Reference

```
#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:

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__)

7.127.1 Macro Definition Documentation

7.127.1.1 #define BEGIN_RCPP

Value:

```
int rcpp_output_type = 0 ;                                                    \
    SEXP rcpp_output_condition = R_NilValue ;                                     \
    try {
```

Definition at line 30 of file macros.h.

Referenced by compileAttributes(), Rcpp::Module::get_class(), class_< Class >::getProperty(), InternalFunction_invoke(), class_< Class >::invoke(), class_< Class >::invoke_notvoid(), class_< Class >::invoke_void(), Module__invoke(), class_< Class >::newInstance(), Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::operator()(), Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::operator()(), PiLeibniz(), class_< Class >::setProperty(), and sourceCppContext().

7.127.1.2 #define END_RCPP VOID_END_RCPP return R_NilValue;

Definition at line 60 of file macros.h.

Referenced by compileAttributes(), Rcpp::Module::get_class(), class_< Class >::getProperty(), InternalFunction_invoke(), class_< Class >::invoke(), class_< Class >::invoke_notvoid(), class_< Class >::invoke_void(), Module__invoke(), class_< Class >::newInstance(), Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::operator()(), Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::operator()(), PiLeibniz(), and sourceCppContext().

7.127.1.3 #define END_RCPP_RETURN_ERROR

Value:

```
}                                                                             \
  catch (Rcpp::internal::InterruptedException &__ex__) {
                \
    return Rcpp::internal::interruptedError();
           \
  }                                                                             \
  catch (std::exception &__ex__) {                                              \
    return exception_to_try_error(__ex__);                                      \
  }                                                                             \
  catch (...) {                                                                 \
    return string_to_try_error("c++ exception (unknown reason)");               \
  }                                                                             \
  return R_NilValue;
```

Definition at line 64 of file macros.h.

7.127.1.4 #define RCPP_DECORATE(*__FUN__*) __FUN__##__rcpp__wrapper__

Definition at line 25 of file macros.h.

7.127.1.5 #define Rcpp_error(*MESSAGE*) throw Rcpp::exception(MESSAGE, __FILE__, __LINE__)

Definition at line 78 of file macros.h.

7.127.1.6 #define RCPP_GET_CLASS(*x*) Rf_getAttrib(x, R_ClassSymbol)

Definition at line 27 of file macros.h.

7.127.1.7 #define RCPP_GET_NAMES(*x*) Rf_getAttrib(x, R_NamesSymbol)

Definition at line 26 of file macros.h.

Referenced by Rcpp::Vector< INTSXP >::containsElementNamed(), Rcpp::Vector< INTSXP >::erase_range__impl(), Rcpp::Vector< INTSXP >::erase_single__impl(), Rcpp::Vector< INTSXP >::findName(), Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get(), Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::get(), Rcpp::Vector< INTSXP >::insert__impl(), Rcpp::Vector< INTSXP >::offset(), Rcpp::Vector< INTSXP >::push_back__impl(), Rcpp::Vector< INTSXP >::push_back_name__impl(), Rcpp::Vector< INTSXP >::push_front__impl(), and Rcpp::Vector< INTSXP >::push_front_name__impl().

7.127.1.8 #define VOID_END_RCPP

Value:

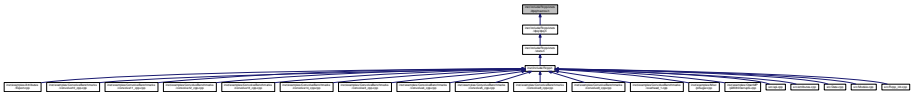
```
}                                                                                                   \
    catch( Rcpp::internal::InterruptedException &__ex__) {
                                \
        rcpp_output_type = 1 ;                                                                      \
    }                                                                                               \
    catch( std::exception& __ex__ ){                                                                \
       rcpp_output_type = 2 ;                                                                       \
       rcpp_output_condition = PROTECT(exception_to_r_condition(__ex__)) ;
                    \
    } catch( ... ){                                                                                 \
       rcpp_output_type = 2 ;                                                                       \
       rcpp_output_condition = PROTECT(string_to_try_error("c++ exception (unknown
       reason)")) ;  \
    }                                                                                               \
    if( rcpp_output_type == 1 ){                                                                    \
       Rf_onintr() ;                                                                                \
    }                                                                                               \
    if( rcpp_output_type == 2 ){                                                                    \
       SEXP stop_sym  = Rf_install( "stop" ) ;                                                      \
       SEXP expr = PROTECT( Rf_lang2( stop_sym , rcpp_output_condition ) ) ;                        \
       Rf_eval( expr, R_GlobalEnv ) ;                                                               \
    }
```

Definition at line 37 of file macros.h.

Referenced by class_< Class >::setProperty().

7.128 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_qIv(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_qIv(p)
- #define R_DT_CIv(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_negInonint(x) (x < 0. || R_D_nonint(x))
- #define R_D_nonint_check(x)

7.128.1 Macro Definition Documentation

7.128.1.1 #define give_log log_p

Definition at line 24 of file macros.h.

7.128.1.2 #define R_D__0 (log_p ? ML_NEGINF : 0.) /* 0 */

Definition at line 27 of file macros.h.

Referenced by Rcpp::stats::d_exp_0(), Rcpp::stats::dgamma_1(), Rcpp::stats::dlnorm_0(), Rcpp::stats::dlnorm_1(), Rcpp::stats::dnorm_0(), Rcpp::stats::dnorm_1(), Rcpp::stats::dunif_0(), and Rcpp::stats::dweibull_1().

7.128.1.3 #define R_D__1 (log_p ? 0. : 1.) /* 1 */

Definition at line 28 of file macros.h.

7.128.1.4 #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.

7.128.1.5 #define R_D_Cval(*p*) (lower_tail ? (0.5 - (p) + 0.5) : (p)) /* 1 - p */

Definition at line 34 of file macros.h.

7.128.1.6 #define R_D_exp(*x*) (log_p ? (x) : ::exp(x)) /* exp(x) */

Definition at line 38 of file macros.h.

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

7.128.1.7 #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.

7.128.1.8 #define R_D_forceint(*x*) floor((x) + 0.5)

Definition at line 117 of file macros.h.

7.128.1.9 #define R_D_LExp(*x*) (log_p ? R_Log1_Exp(x) : ::log1p(-x))

Definition at line 46 of file macros.h.

7.128.1.10 #define R_D_log(*p*) (log_p ? (p) : ::log(p)) /* log(p) */

Definition at line 39 of file macros.h.

7.128.1.11 #define R_D_Lval(p) (lower_tail ? (p) : (0.5 - (p) + 0.5)) /* p */

Definition at line 33 of file macros.h.

7.128.1.12 #define R_D_negInonint(x) (x < 0. || R_D_nonint(x))

Definition at line 120 of file macros.h.

7.128.1.13 #define R_D_nonint(x) (fabs((x) - floor((x)+0.5)) > 1e-7)

Definition at line 118 of file macros.h.

7.128.1.14 #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.

7.128.1.15 #define R_D_qIv(p) (log_p ? ::exp(p) : (p)) /* p in qF(p,..) */

Definition at line 37 of file macros.h.

7.128.1.16 #define R_D_val(x) (log_p ? ::log(x) : (x)) /* x in pF(x,..) */

Definition at line 36 of file macros.h.

Referenced by Rcpp::stats::punif_0().

7.128.1.17 #define R_DT_0 (lower_tail ? R_D__0 : R_D__1) /* 0 */

Definition at line 29 of file macros.h.

Referenced by Rcpp::stats::p_exp_0(), Rcpp::stats::plnorm_0(), Rcpp::stats::plnorm_1(), Rcpp::stats::pnorm_0(), Rcpp::stats::pnorm_1(), Rcpp::stats::punif_0(), Rcpp::stats::pweibull_1(), and Rcpp::stats::q_exp_0().

7.128.1.18 #define R_DT_1 (lower_tail ? R_D__1 : R_D__0) /* 1 */

Definition at line 30 of file macros.h.

Referenced by Rcpp::stats::pnorm_0(), Rcpp::stats::pnorm_1(), and Rcpp::stats::punif_0().

7.128.1.19 #define R_DT_Cexp(x) R_D_exp(R_D_Cval(x)) /* exp(1 - x) */

Definition at line 60 of file macros.h.

7.128.1.20 #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.

7.128.1.21 #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.

Referenced by Rcpp::stats::q_exp_0(), and Rcpp::stats::qweibull_1().

7.128.1.22 #define R_DT_Cval(x) (lower_tail ? R_D_Clog(x) : R_D_val(x))

Definition at line 49 of file macros.h.

7.128.1.23 #define R_DT_exp(x) R_D_exp(R_D_Lval(x)) /* exp(x) */

Definition at line 59 of file macros.h.

7.128.1.24 #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.

7.128.1.25 #define R_DT_Log(p) (lower_tail? (p) : R_Log1_Exp(p))

Definition at line 64 of file macros.h.

7.128.1.26 #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.

Referenced by Rcpp::stats::qunif_0(), and Rcpp::stats::qunif_1().

7.128.1.27 #define R_DT_val(x) (lower_tail ? R_D_val(x) : R_D_Clog(x))

Definition at line 48 of file macros.h.

7.128.1.28 #define R_Log1_Exp(x) ((x) > -M_LN2 ? ::log(-::expm1(x)) : ::log1p(-::exp(x)))

Definition at line 43 of file macros.h.

7.128.1.29 #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.

7.128.1.30 #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.

Referenced by Rcpp::stats::plogis_0(), and Rcpp::stats::plogis_1().

7.128.1.31 #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.

Referenced by Rcpp::stats::qlnorm_0(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_0(), Rcpp::stats::qlogis_1(), and Rcpp::stats::qweibull_1().

7.128.1.32 #define R_Q_P01_check(*p*)

Value:

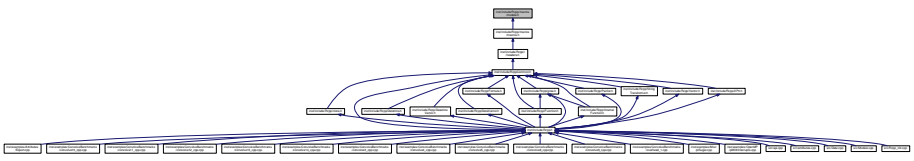
```
if ((log_p      && p > 0) ||                    \
        (!log_p && (p < 0 || p > 1)) )          \
        return R_NaN
```

Definition at line 68 of file macros.h.

Referenced by Rcpp::stats::qunif_0(), and Rcpp::stats::qunif_1().

7.129 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)

7.129.1 Macro Definition Documentation

7.129.1.1 #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.

7.129.1.2 #define RCPP_EXPOSED_CLASS(*CLASS*)

Value:

```
class CLASS;                    \
  RCPP_EXPOSED_CLASS_NODECL(CLASS)
```

Definition at line 69 of file module.h.

7.129.1.3 #define RCPP_EXPOSED_CLASS_NODECL(*CLASS*)

Value:

```
RCPP_EXPOSED_AS(CLASS)          \
  RCPP_EXPOSED_WRAP(CLASS)
```

Definition at line 65 of file module.h.

7.129.1.4 #define RCPP_EXPOSED_ENUM(*CLASS*)

Value:

```
class CLASS;                    \
  RCPP_EXPOSED_ENUM_NODECL(CLASS)
```

Definition at line 83 of file module.h.

7.129.1.5 #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.

7.129.1.6 #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.

7.129.1.7 #define RCPP_EXPOSED_ENUM_WRAP(*CLASS*) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS >{typedef wrap_type_enum_tag wrap_category ; } ; }}

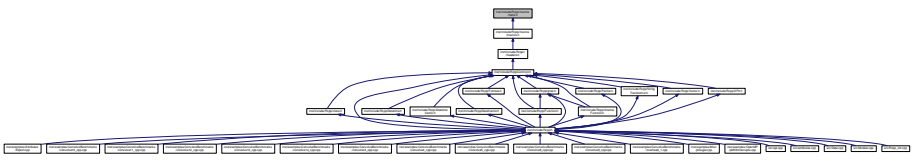
Definition at line 77 of file module.h.

7.129.1.8 #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.

7.130 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)

7.130.1 Macro Definition Documentation

7.130.1.1 #define RCPP_ENUM_TRAITS(*__ENUM__*) RCPP_TRAITS(__ENUM__,INTSXP)

Definition at line 40 of file traits.h.

7.130.1.2 #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.

7.131 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 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

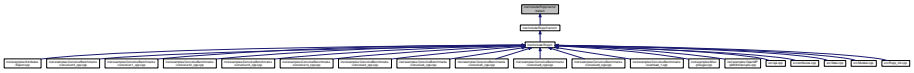
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.132 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

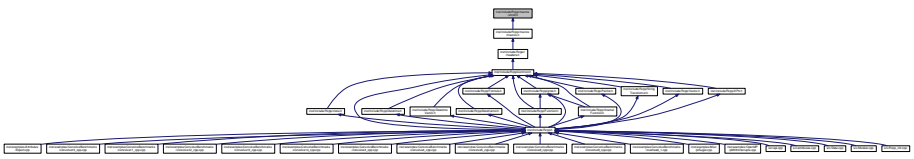
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.133 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)

7.133.1 Macro Definition Documentation

7.133.1.1 #define RCPP_LOOP_UNROLL(*TARGET, SOURCE*)

Value:

```
int __trip_count = n >> 2 ;                       \
int 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++ ;               \
  case 2:                                         \
      TARGET[i] = SOURCE[i] ; i++ ;               \
  case 1:                                         \
      TARGET[i] = SOURCE[i] ; i++ ;               \
  case 0:                                         \
  default:                                        \
      {}                                          \
}
```

Definition at line 47 of file unroll.h.

Referenced by Rcpp::Vector< INTSXP >::import_expression(), and Rcpp::MatrixColumn< RTYPE >::operator=().

7.133.1.2 #define RCPP_LOOP_UNROLL_LHSFUN(*TARGET, FUN, SOURCE*)

Value:

```
int __trip_count = n >> 2 ;                       \
int 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++ ;              \
  case 2:                                         \
      TARGET[FUN(i)] = SOURCE[i] ; i++ ;              \
  case 1:                                         \
      TARGET[FUN(i)] = SOURCE[i] ; i++ ;              \
  case 0:                                         \
  default:                                        \
      {}                                          \
}
```

Definition at line 68 of file unroll.h.

Referenced by Rcpp::MatrixRow< RTYPE >::operator=().

7.133.1.3 #define RCPP_LOOP_UNROLL_PTR(*TARGET, SOURCE*)

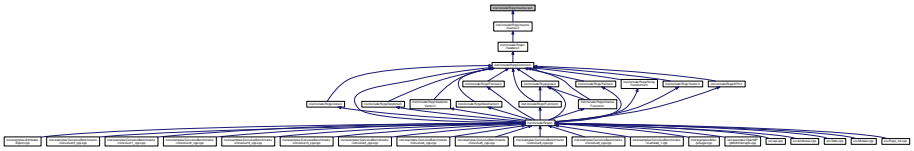
Value:

```
int __trip_count = n >> 2 ;                     \
int 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.

7.134 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__)

7.134.1 Macro Definition Documentation

7.134.1.1 #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.

7.134.1.2 #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.

7.134.1.3 #define RCPP_XP_FIELD_SET(__NAME__, __CLASS__, __FIELD__)

Value:

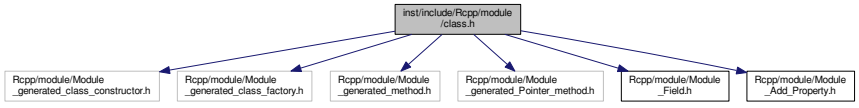
```
extern "C" SEXP RCPP_PP_CAT(__NAME__,__rcpp_info__)(){          \
        using Rcpp::_ ;                                             \
        Rcpp::List info = Rcpp::List::create(                       \
        _["class"]  = #__CLASS__  ,                                 \
        _["field"]  = #__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.

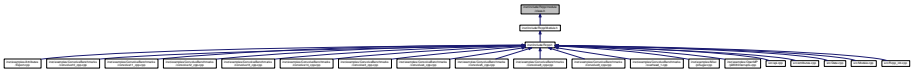
7.135 inst/include/Rcpp/module/class.h File Reference

```
#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:

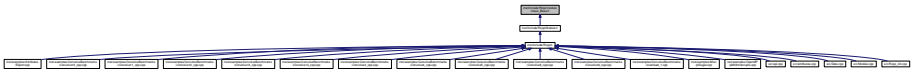


Classes

- class class_< Class >

7.136 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

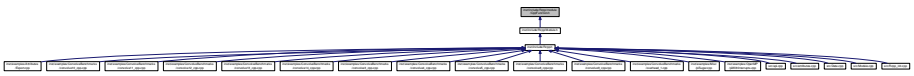
Namespaces

- Rcpp

  Rcpp API.

7.137 inst/include/Rcpp/module/CppFunction.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

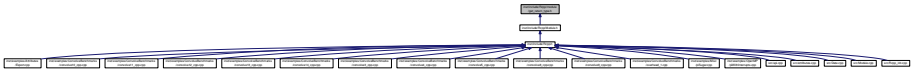
- class Rcpp::CppFunctionBase
- class Rcpp::CppFunction

Namespaces

- Rcpp
  Rcpp API.

7.138 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

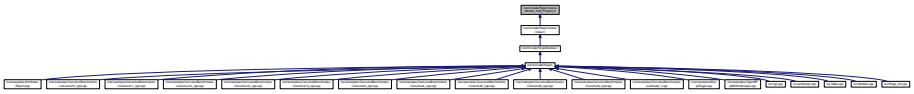
- Rcpp
  Rcpp API.

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 > ()

7.139 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)

7.139.1 Function Documentation

7.139.1.1 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.

Referenced by property().

7.139.1.2 template<typename PROP > self& property (const char * *name_*, PROP(*)(Class *) *GetMethod*, const char * *docstring*)

Definition at line 38 of file Module_Add_Property.h.

7.139.1.3 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.

References property().

Here is the call graph for this function:

7.139.1.4 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.

References property().

Here is the call graph for this function:

7.139.1.5 template<typename PROP > self& property (const char * *name_*, PROP(*)(Class *) *GetMethod*, void(Class::*)(PROP) *SetMethod*, const char * *docstring* = 0)

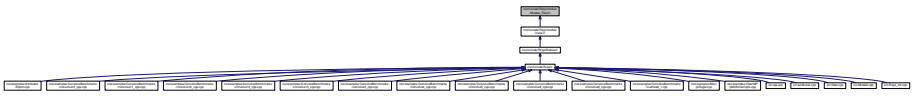
Definition at line 81 of file Module_Add_Property.h.

7.139.1.6 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.

7.140 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)

7.140.1 Function Documentation

7.140.1.1 template<typename T > self& field (const char * *name_*, T Class::* *ptr*, const char * *docstring* = 0)

Definition at line 68 of file Module_Field.h.

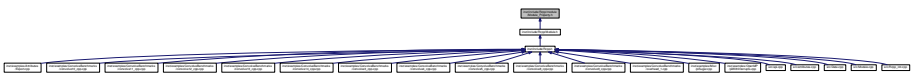
Referenced by Rcpp::S4_CppConstructor< Class >::S4_CppConstructor(), Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods(), and Rcpp::S4_field< Class >::S4_field().

7.140.1.2 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.

7.141 inst/include/Rcpp/module/Module_Property.h File Reference

This graph shows which files directly or indirectly include this file:

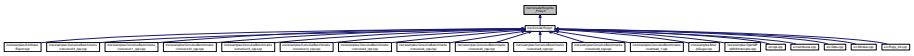


Classes

- 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 >

7.142 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

7.143 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

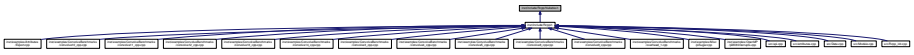
- Argument Rcpp::Named (const std::string &name)
- template<typename T >
  traits::named_object< T > Rcpp::Named (const std::string &name, const T &o)

Variables

- static internal::NamedPlaceHolder Rcpp::_

7.144 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

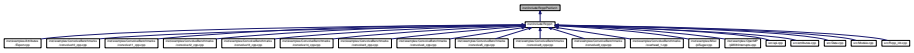
  Rcpp API.

7.145 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)

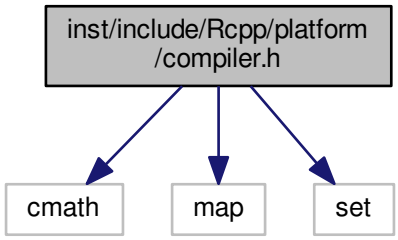
Variables

- public Rcpp::DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< Pairlist_Impl< StoragePolicy > >

7.146 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

7.146.1 Macro Definition Documentation

7.146.1.1 #define RCPP_UNORDERED_MAP std::map

Definition at line 178 of file compiler.h.

7.146.1.2 #define RCPP_UNORDERED_SET std::set

Definition at line 187 of file compiler.h.

7.146.1.3 #define RCPP_USING_MAP

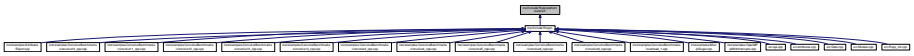
Definition at line 177 of file compiler.h.

7.146.1.4 #define RCPP_USING_SET

Definition at line 186 of file compiler.h.

7.147 inst/include/Rcpp/platform/solaris.h File Reference

This graph shows which files directly or indirectly include this file:



7.148 inst/include/Rcpp/print.h File Reference

This graph shows which files directly or indirectly include this file:

Namespaces

- Rcpp

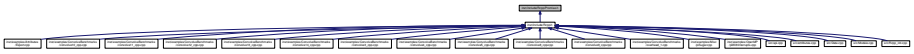
  Rcpp API.

Functions

- void Rcpp::print (SEXP s)
- void Rcpp::warningcall (SEXP call, const std::string &s)

7.149 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)

7.150 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.

7.151 inst/include/Rcpp/protection/Shelter.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- class Rcpp::Shelter< T >

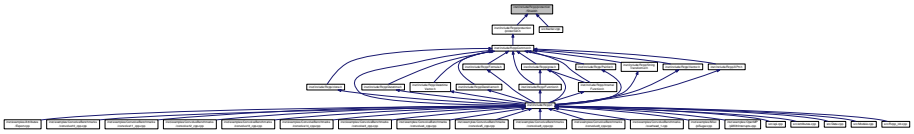
Namespaces

- Rcpp

  Rcpp API.

7.152 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)

7.153 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.

7.154 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.

7.155 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.

7.156 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.

7.157 inst/include/Rcpp/proxy/GenericProxy.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct Rcpp::GenericProxy< Proxy >

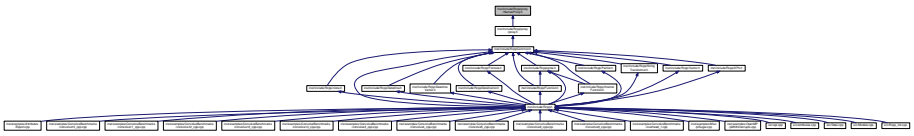
Namespaces

- Rcpp

  Rcpp API.

7.158 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.

7.159 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)

7.160 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.

7.161 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.

7.162 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.

7.163 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 NORET

7.163.1 Macro Definition Documentation

7.163.1.1 #define MAXELTSIZE 8192

Definition at line 25 of file headers.h.

7.163.1.2 #define NORET

Definition at line 64 of file headers.h.

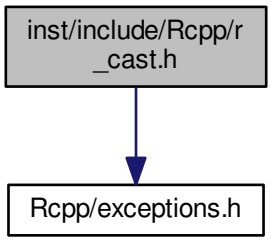
7.163.1.3 #define R_NO_REMAP

Definition at line 26 of file headers.h.

7.164 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

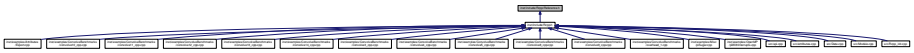
Functions

- 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)

7.165 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 > >

7.166 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)

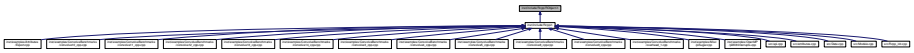
- double R::rchisq (double df)
- double R::dnchisq (double x, double df, double ncp, int lg)
- double R::pnchisq (double x, double df, double ncp, int lt, int lg)
- double R::qnchisq (double p, double df, double ncp, int lt, int lg)
- double R::rnchisq (double df, double lb)
- double R::df (double x, 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 al, double ex, double *bk)
- double R::bessel_y_ex (double x, double al, double *by)
- double R::hypot (double a, double b)
- double R::pythag (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)

7.167 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)

7.168 inst/include/Rcpp/routines.h File Reference

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 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 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)
- attribute_hidden SEXP get_string_elt (SEXP s, int i)
- attribute_hidden const char * char_get_string_elt (SEXP s, int i)
- attribute_hidden void set_string_elt (SEXP s, int i, SEXP v)
- attribute_hidden void char_set_string_elt (SEXP s, int i, const char *v)
- attribute_hidden SEXP * get_string_ptr (SEXP s)
- attribute_hidden SEXP get_vector_elt (SEXP v, int i)
- attribute_hidden void set_vector_elt (SEXP v, int 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 ()

7.168.1 Macro Definition Documentation

7.168.1.1 #define GET_CALLABLE(__FUN__) (Fun) R_GetCCallable("Rcpp", __FUN__)

Definition at line 68 of file routines.h.

Referenced by char_get_string_elt(), char_nocheck(), char_set_string_elt(), dataptr(), demangle(), 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_get_current_error(), rcpp_get_stack_trace(), rcpp_set_stack_trace(), reset_current_error(), set_string_elt(), set_vector_elt(), setCurrentScope(), short_file_name(), stack_trace(), and Rcpp::type2name().

7.168.2 Function Documentation

7.168.2.1 attribute_hidden const char* char_get_string_elt (SEXP *s*, int *i*) [inline]

Definition at line 158 of file routines.h.

References GET_CALLABLE.

7.168.2.2 attribute_hidden const char* char_nocheck (SEXP *x*) [inline]

Definition at line 200 of file routines.h.

References GET_CALLABLE.

7.168.2.3 attribute_hidden void char_set_string_elt (SEXP *s*, int *i*, const char * *v*) [inline]

Definition at line 170 of file routines.h.

References GET_CALLABLE.

7.168.2.4 attribute_hidden void* dataptr (SEXP *x*) [inline]

Definition at line 206 of file routines.h.

References GET_CALLABLE.

7.168.2.5 attribute_hidden std::string demangle (const std::string & *name*) [inline]

Definition at line 134 of file routines.h.

References GET_CALLABLE.

7.168.2.6 attribute_hidden int error_occured ( ) [inline]

Definition at line 236 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.168.2.7 attribute_hidden int* get_cache (int *n*) [inline]

Definition at line 224 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and registerFunctions().

7.168.2.8 attribute_hidden SEXP get_string_elt (SEXP *s*, int *i*) [inline]

Definition at line 152 of file routines.h.

References GET_CALLABLE.

7.168.2.9 attribute_hidden SEXP* get_string_ptr (SEXP *s*) [inline]

Definition at line 176 of file routines.h.

References GET_CALLABLE.

7.168.2.10 attribute_hidden SEXP get_vector_elt (SEXP *v*, int *i*) [inline]

Definition at line 182 of file routines.h.

References GET_CALLABLE.

7.168.2.11 attribute_hidden SEXP* get_vector_ptr (SEXP *v*) [inline]

Definition at line 194 of file routines.h.

References GET_CALLABLE.

7.168.2.12 attribute_hidden Rcpp::Module* getCurrentScope ( ) [inline]

Definition at line 212 of file routines.h.

References GET_CALLABLE.

Referenced by class_< Class >::derives(), class_< Class >::get_instance(), registerFunctions(), and Rcpp::enum_< Enum, Parent >::~enum_().

7.168.2.13 attribute_hidden SEXP rcpp_get_current_error ( ) [inline]

Definition at line 242 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.168.2.14 attribute_hidden SEXP rcpp_get_stack_trace ( ) [inline]

Definition at line 122 of file routines.h.

References GET_CALLABLE.

Referenced by exception_to_r_condition(), and registerFunctions().

7.168.2.15 attribute_hidden SEXP rcpp_set_stack_trace (SEXP *e*) [inline]

Definition at line 128 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::exception::exception(), exception_to_r_condition(), and registerFunctions().

7.168.2.16 attribute_hidden SEXP reset_current_error ( ) [inline]

Definition at line 230 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.168.2.17 attribute_hidden void set_string_elt (SEXP *s*, int *i*, SEXP *v*) [inline]

Definition at line 164 of file routines.h.

References GET_CALLABLE.

7.168.2.18 attribute_hidden void set_vector_elt (SEXP *v*, int *i*, SEXP *x*) [inline]

Definition at line 188 of file routines.h.

References GET_CALLABLE.

7.168.2.19 attribute_hidden void setCurrentScope (Rcpp::Module * *mod*) [inline]

Definition at line 218 of file routines.h.

References GET_CALLABLE.

Referenced by R_init_Rcpp(), and registerFunctions().

7.168.2.20 attribute_hidden const char* short_file_name (const char * *file*) [inline]

Definition at line 140 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.168.2.21 attribute_hidden SEXP stack_trace (const char * *file,* int *line*) [inline]

Definition at line 146 of file routines.h.

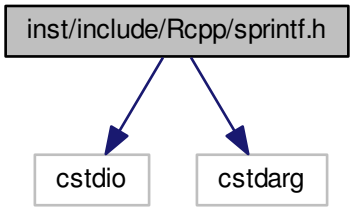
References GET_CALLABLE.

Referenced by Rcpp::exception::exception(), and registerFunctions().

7.169 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

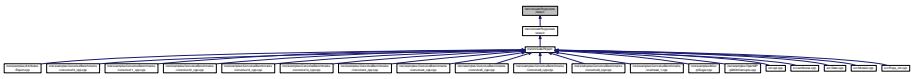
- Rcpp
  Rcpp API.

Functions

- template<int MAX_SIZE>
  std::string Rcpp::sprintf (const char *format,...)

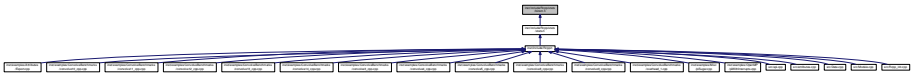
7.170 inst/include/Rcpp/stats/beta.h File Reference

This graph shows which files directly or indirectly include this file:



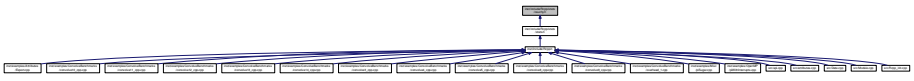
7.171 inst/include/Rcpp/stats/binom.h File Reference

This graph shows which files directly or indirectly include this file:



7.172 inst/include/Rcpp/stats/cauchy.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

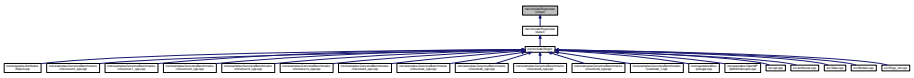
- Rcpp
  Rcpp API.
- Rcpp::stats

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)

7.173 inst/include/Rcpp/stats/chisq.h File Reference

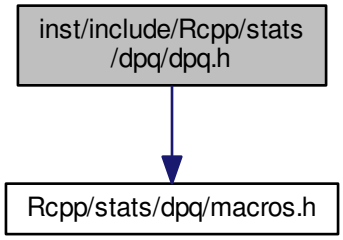
This graph shows which files directly or indirectly include this file:



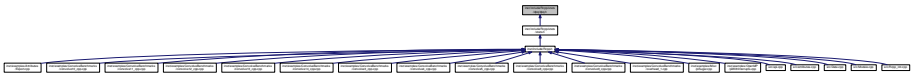
7.174 inst/include/Rcpp/stats/dpq/dpq.h File Reference

```
#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 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

Macros

- #define RCPP_DPQ_0(__NAME__, __D__, __P__, __Q__)
- #define RCPP_DPQ_1(__NAME__, __D__, __P__, __Q__)
- #define RCPP_DPQ_2(__NAME__, __D__, __P__, __Q__)
- #define RCPP_DPQ_3(__NAME__, __D__, __P__, __Q__)

7.174.1 Macro Definition Documentation

7.174.1.1 #define RCPP_DPQ_0(*__NAME__*, *__D__*, *__P__*, *__Q__*)

Value:

```
namespace Rcpp {                                                                \
template <int RTYPE, bool NA, typename T>                                       \
inline stats::D0<RTYPE,NA,T> d##__NAME__(                                       \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, bool log = false
                \
) {                                                                             \
    return stats::D0<RTYPE,NA,T>( __D__, x, log );                              \
}                                                                               \
template <int RTYPE, bool NA, typename T>                                       \
inline stats::P0<RTYPE,NA,T> p##__NAME__(                                       \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, bool lower = true, bool
      log = false         \
) {                                                                             \
    return stats::P0<RTYPE,NA,T>( __P__, x, lower, log );                       \
}                                                                               \
template <int RTYPE, bool NA, typename T>                                       \
inline stats::Q0<RTYPE,NA,T> q##__NAME__(                                       \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, bool lower = true, bool
      log = false         \
) {                                                                             \
    return stats::Q0<RTYPE,NA,T>( __Q__, x, lower, log );                       \
} }
```

Definition at line 318 of file dpq.h.

Referenced by Rcpp::stats::p_exp_0(), Rcpp::stats::qcauchy_1(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_1(), Rcpp::stats::qnorm_0(), and Rcpp::stats::qunif_0().

7.174.1.2 #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.

Referenced by Rcpp::stats::qcauchy_1(), Rcpp::stats::qgamma_1(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_1(), Rcpp::stats::qnorm_0(), Rcpp::stats::qunif_0(), and Rcpp::stats::qweibull_1().

7.174.1.3 #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.

Referenced by Rcpp::stats::qcauchy_1(), Rcpp::stats::qgamma_1(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_1(), Rcpp::stats::qnorm_0(), Rcpp::stats::qunif_0(), and Rcpp::stats::qweibull_1().

7.174.1.4 #define RCPP_DPQ_3(__NAME__, __D__, __P__, __Q__)

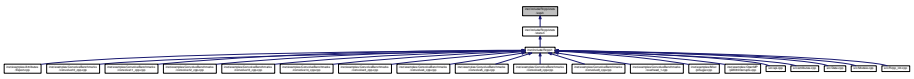
Value:

```
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.

7.175 inst/include/Rcpp/stats/exp.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

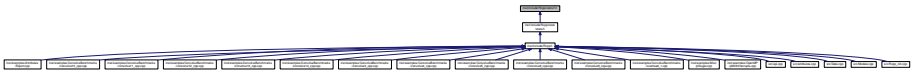
- Rcpp
  Rcpp API.
- Rcpp::stats

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)

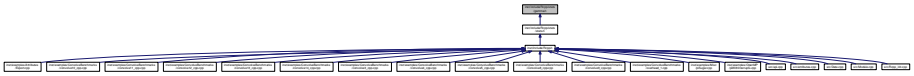
7.176 inst/include/Rcpp/stats/f.h File Reference

This graph shows which files directly or indirectly include this file:



7.177 inst/include/Rcpp/stats/gamma.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

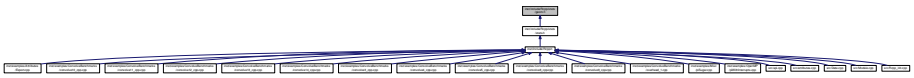
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dgamma_1 (double x, double shape, int log_p)
- double Rcpp::stats::pgamma_1 (double x, double alph, int lower_tail, int log_p)
- double Rcpp::stats::qgamma_1 (double p, double alpha, int lower_tail, int log_p)

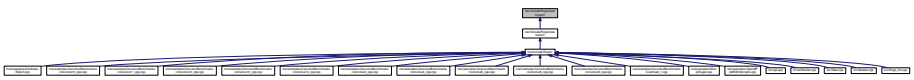
7.178 inst/include/Rcpp/stats/geom.h File Reference

This graph shows which files directly or indirectly include this file:



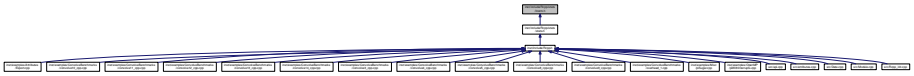
7.179 inst/include/Rcpp/stats/hyper.h File Reference

This graph shows which files directly or indirectly include this file:



7.180 inst/include/Rcpp/stats/lnorm.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

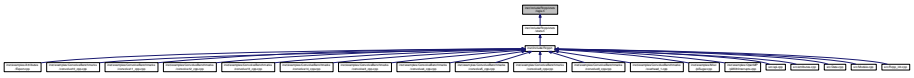
  Rcpp API.
- Rcpp::stats

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)

7.181 inst/include/Rcpp/stats/logis.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

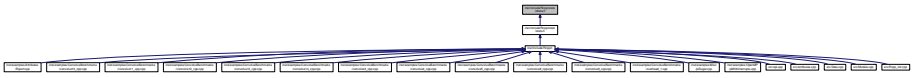
  Rcpp API.
- Rcpp::stats

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)

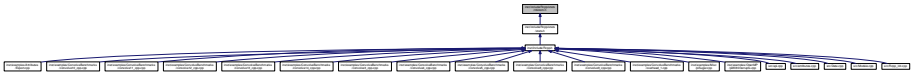
7.182 inst/include/Rcpp/stats/nbeta.h File Reference

This graph shows which files directly or indirectly include this file:



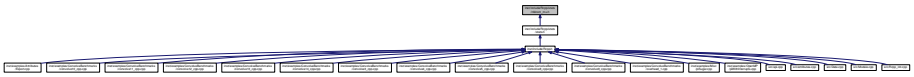
7.183 inst/include/Rcpp/stats/nbinom.h File Reference

This graph shows which files directly or indirectly include this file:



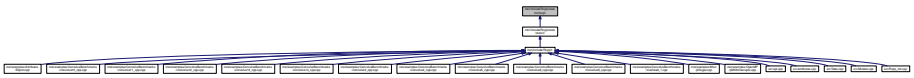
7.184 inst/include/Rcpp/stats/nbinom_mu.h File Reference

This graph shows which files directly or indirectly include this file:



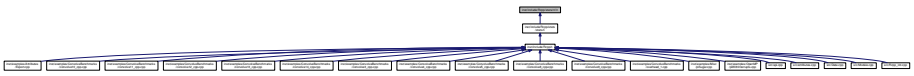
7.185 inst/include/Rcpp/stats/nchisq.h File Reference

This graph shows which files directly or indirectly include this file:



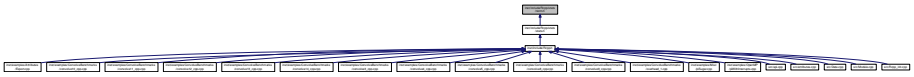
7.186 inst/include/Rcpp/stats/nf.h File Reference

This graph shows which files directly or indirectly include this file:



7.187 inst/include/Rcpp/stats/norm.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

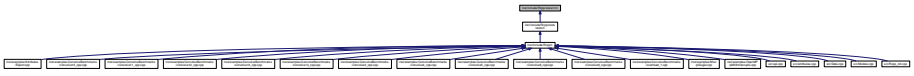
  Rcpp API.
- Rcpp::stats

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)

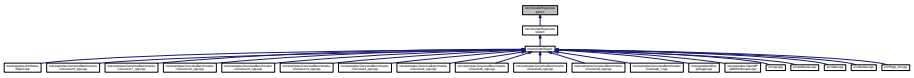
7.188 inst/include/Rcpp/stats/nt.h File Reference

This graph shows which files directly or indirectly include this file:



7.189 inst/include/Rcpp/stats/pois.h File Reference

This graph shows which files directly or indirectly include this file:

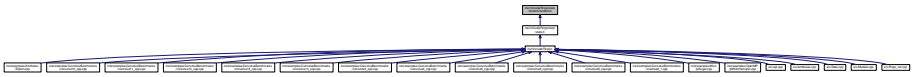


7.190 inst/include/Rcpp/stats/random/random.h File Reference

```
#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::RNGScope
- class Rcpp::Generator< T >

Namespaces

- Rcpp

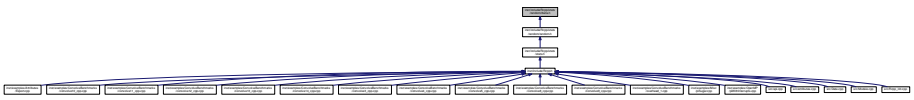
  Rcpp API.

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)

7.191 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

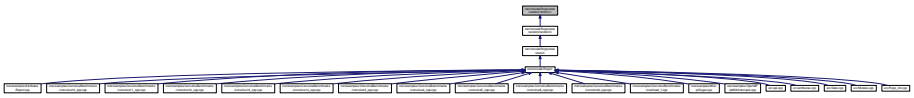
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.192 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

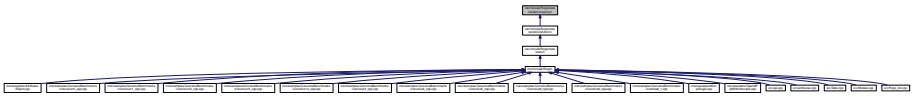
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.193 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

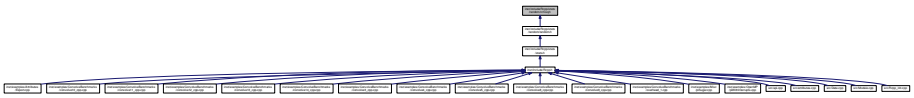
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.194 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

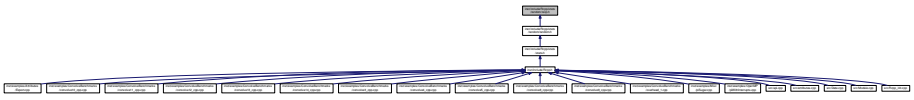
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.195 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

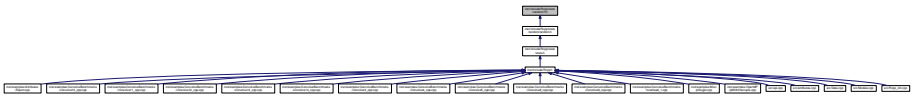
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.196 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

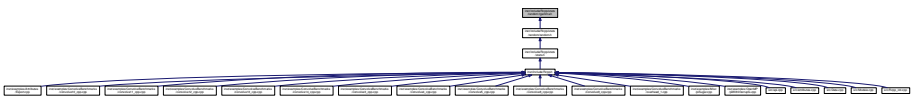
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.197 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

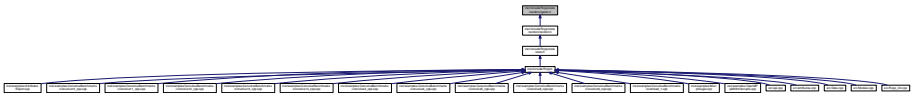
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.198 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

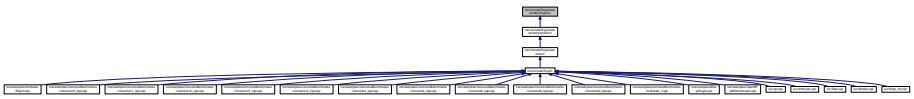
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.199 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

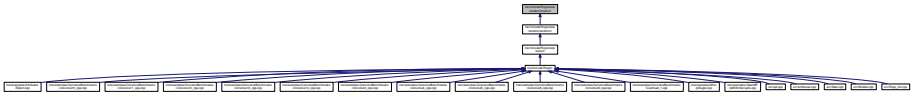
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.200 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

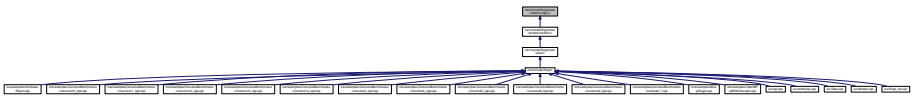
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.201 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

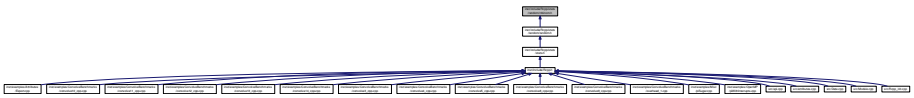
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.202 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

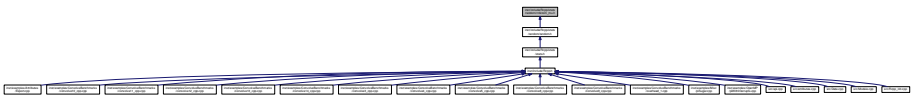
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.203 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

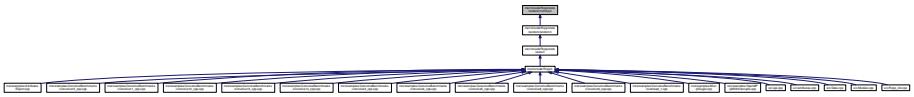
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.204 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

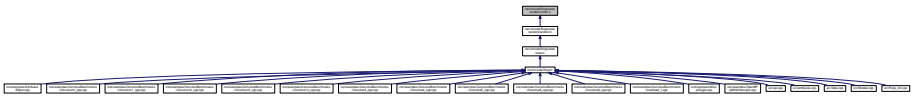
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.205 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

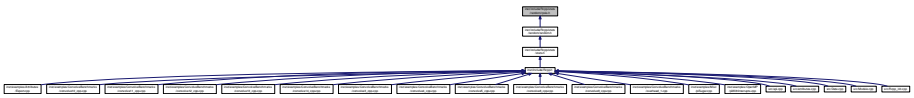
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.206 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

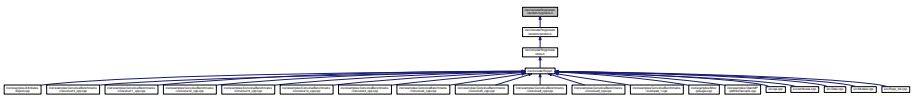
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.207 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

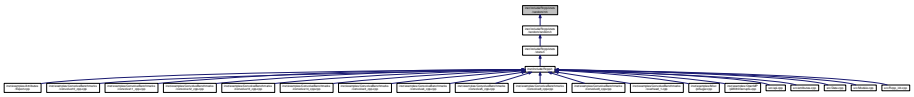
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.208 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

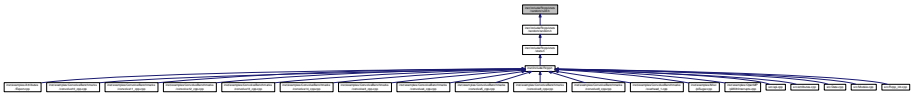
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.209 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

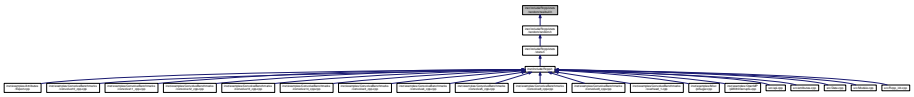
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.210 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

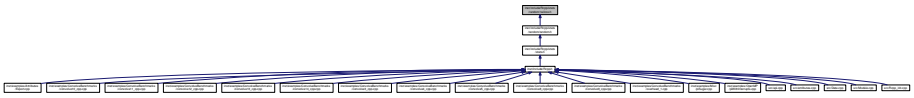
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.211 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

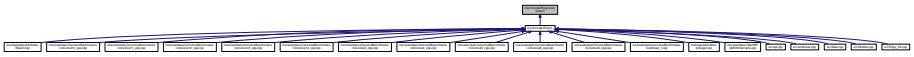
  Rcpp API.
- Rcpp::stats

7.212 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

7.212.1 Macro Definition Documentation

7.212.1.1 #define ML_NAN R_NaN

Definition at line 29 of file stats.h.

Referenced by Rcpp::stats::dnorm_1(), and Rcpp::stats::pnorm_1().

7.212.1.2 #define ML_NEGINF R_NegInf

Definition at line 28 of file stats.h.

Referenced by Rcpp::stats::qlogis_0(), and Rcpp::stats::qlogis_1().

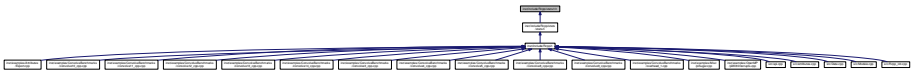
7.212.1.3 #define ML_POSINF R_PosInf

Definition at line 27 of file stats.h.

Referenced by Rcpp::stats::dgamma_1(), Rcpp::stats::dweibull_1(), Rcpp::stats::qlnorm_0(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_0(), Rcpp::stats::qlogis_1(), and Rcpp::stats::qweibull_1().

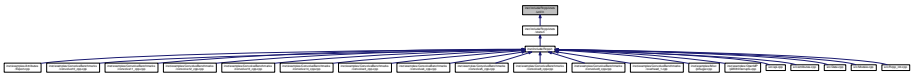
7.213 inst/include/Rcpp/stats/t.h File Reference

This graph shows which files directly or indirectly include this file:



7.214 inst/include/Rcpp/stats/unif.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

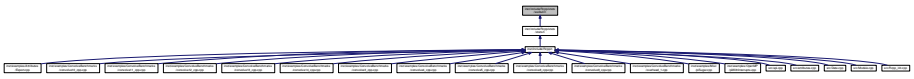
  Rcpp API.
- Rcpp::stats

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)

7.215 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)

7.216 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.

7.217 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

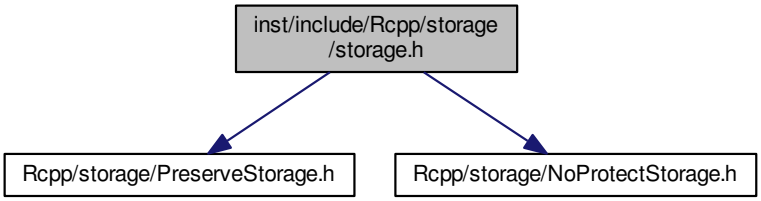
- Rcpp

  Rcpp API.

7.218 inst/include/Rcpp/storage/storage.h File Reference

```
#include <Rcpp/storage/PreserveStorage.h>
#include <Rcpp/storage/NoProtectStorage.h>
```

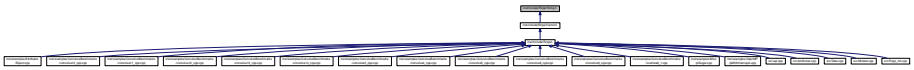
Include dependency graph for storage.h:



This graph shows which files directly or indirectly include this file:

7.219 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)

7.219.1 Macro Definition Documentation

7.219.1.1 #define RCPP_STRING_DEBUG(*MSG*)

Definition at line 37 of file String.h.

Referenced by Rcpp::String::operator+=(), Rcpp::String::setData(), Rcpp::String::String(), and Rcpp::wrap< Rcpp::↩ String >().

7.219.1.2 #define RCPP_STRING_DEBUG_1(*fmt, MSG*)

Definition at line 38 of file String.h.

Referenced by Rcpp::String::append_wide_string(), and Rcpp::String::get_sexp().

7.219.1.3 #define RCPP_STRING_DEBUG_2(*fmt, M1, M2*)

Definition at line 39 of file String.h.

Referenced by Rcpp::String::replace_all(), Rcpp::String::replace_first(), and Rcpp::String::replace_last().

7.219.1.4 #define RCPP_STRING_DEBUG_3(*fmt, M1, M2, M3*)

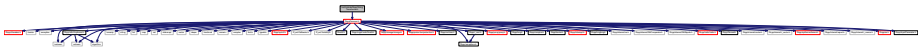
Definition at line 40 of file String.h.

7.219.1.5 #define RCPP_STRING_DEBUG_LEVEL 0

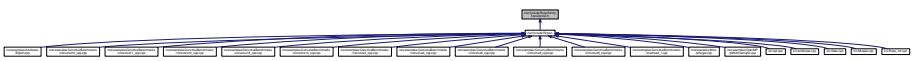
Definition at line 26 of file String.h.

7.220 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.

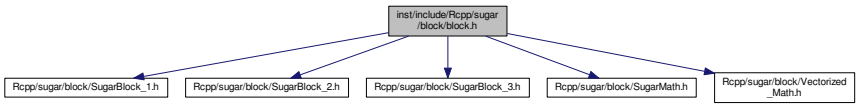
Functions

- template<typename UnaryOperator >
  StringTransformer< UnaryOperator > Rcpp::make_string_transformer (const UnaryOperator &fun)

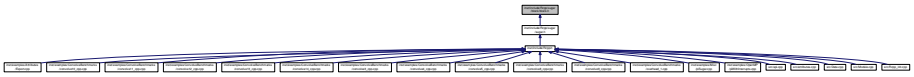
7.221 inst/include/Rcpp/sugar/block/block.h File Reference

```
#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:

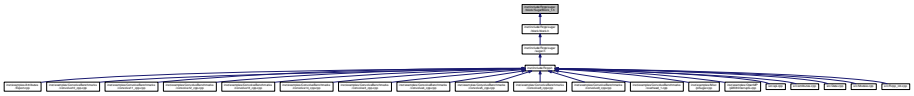


This graph shows which files directly or indirectly include this file:



7.222 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__)

7.222.1 Macro Definition Documentation

7.222.1.1 #define SB1_T VectorBase<REALSXP,NA,T>

Definition at line 47 of file SugarBlock_1.h.

7.222.1.2 #define SUGAR_BLOCK_1(__NAME__, __SYMBOL__)

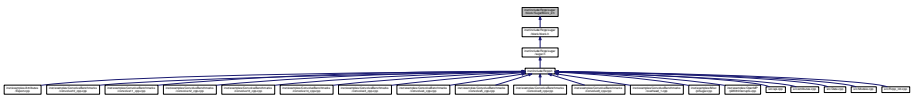
Value:

```
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.

7.223 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__)

7.223.1 Macro Definition Documentation

7.223.1.1 #define SB2_LHT VectorBase<REALSXP,LHS_NA,LHS_T>

Definition at line 89 of file SugarBlock_2.h.

7.223.1.2 #define SB2_RHT VectorBase<REALSXP,RHS_NA,RHS_T>

Definition at line 90 of file SugarBlock_2.h.

7.223.1.3 #define SUGAR_BLOCK_2(*__NAME__*, *__SYMBOL__*)

Definition at line 92 of file SugarBlock_2.h.

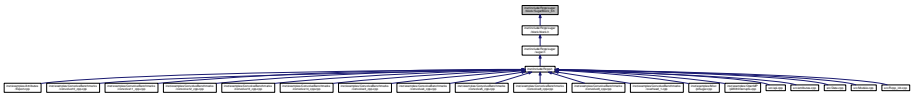
Referenced by Rcpp::internal::lfactorial().

7.223.1.4 #define SUGAR_BLOCK_2_NA(*__NAME__*, *__SYMBOL__*, *__NA__*)

Definition at line 127 of file SugarBlock_2.h.

7.224 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__)

7.224.1 Macro Definition Documentation

7.224.1.1 #define SB3_T1 VectorBase<REALSXP,T1_NA,T1>

Definition at line 97 of file SugarBlock_3.h.

7.224.1.2 #define SB3_T2 VectorBase<REALSXP,T2_NA,T2>

Definition at line 98 of file SugarBlock_3.h.

7.224.1.3 #define SB3_T3 VectorBase<REALSXP,T3_NA,T3>

Definition at line 99 of file SugarBlock_3.h.

7.224.1.4 #define SUGAR_BLOCK_3(__NAME__, __SYMBOL__)

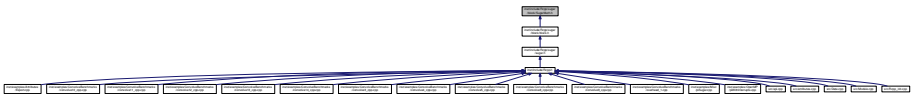
Value:

```
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.

7.225 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__)

7.225.1 Macro Definition Documentation

7.225.1.1 #define SUGAR_MATH_1(*__NAME__*, *__SYMBOL__*)

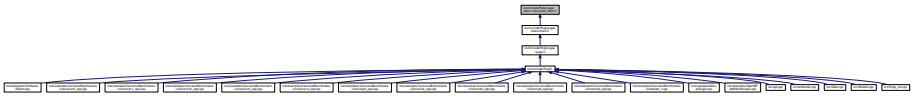
Value:

```
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.

7.226 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)

7.226.1 Macro Definition Documentation

7.226.1.1 #define VECTORIZED_MATH_1(*__NAME__*, *__SYMBOL__*)

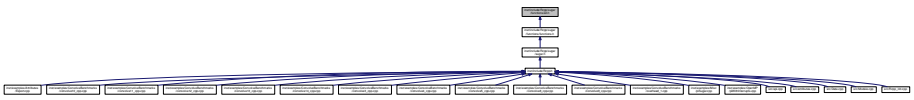
Value:

```
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.

7.227 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

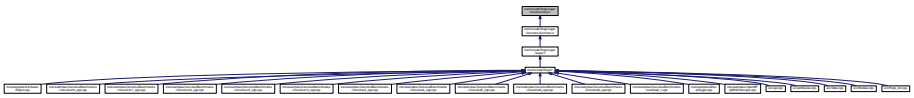
  Rcpp API.
- Rcpp::sugar

Functions

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

7.228 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

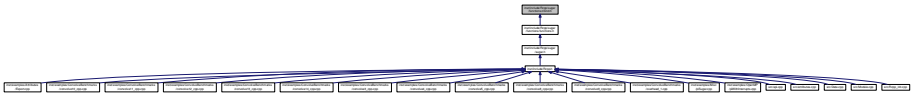
  Rcpp API.
- Rcpp::sugar

Functions

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

7.229 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< typename scalar< RTYPE >::type > >
- 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::MakeBindable (const T &t)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)

- 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)
- 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)
- 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)
- 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)

7.229.1 Macro Definition Documentation

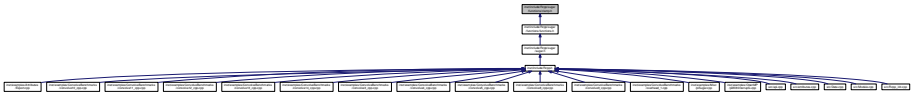
7.229.1.1 #define MakeBindable(*x*) (cbind_impl::detail::MakeBindable(x)(x))

Definition at line 510 of file cbind.h.

Referenced by Rcpp::sugar::cbind().

7.230 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

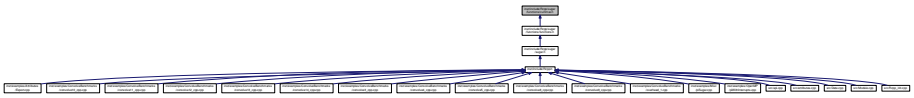
  Rcpp API.
- Rcpp::sugar

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)

7.231 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

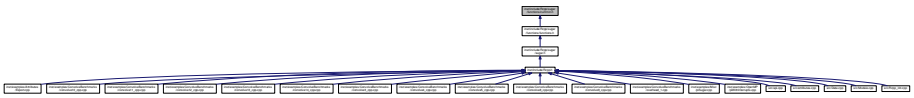
  Rcpp API.
- Rcpp::sugar

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)

7.232 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

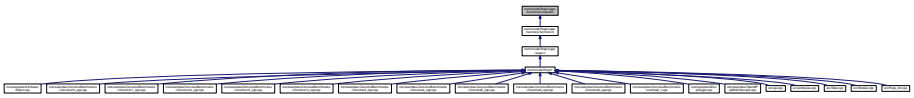
  Rcpp API.
- Rcpp::sugar

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)

7.233 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

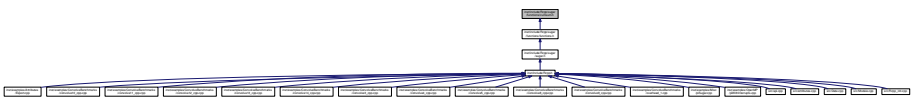
- Rcpp
  Rcpp API.
- Rcpp::sugar

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)

7.234 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

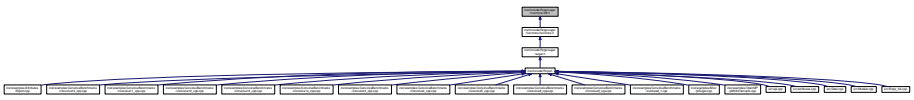
- Rcpp
  Rcpp API.
- Rcpp::sugar

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)

7.235 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

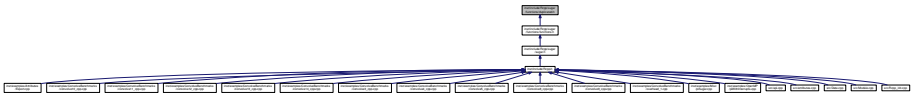
- Rcpp
  Rcpp API.
- Rcpp::sugar

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)

7.236 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

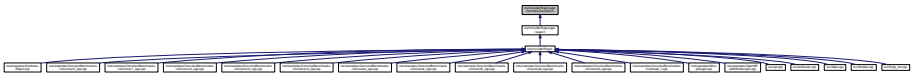
- template<int RTYPE, bool NA, typename T >
  LogicalVector Rcpp::duplicated (const VectorBase< RTYPE, NA, T > &x)

7.237 inst/include/Rcpp/sugar/functions/functions.h File Reference

```
#include <cmath>
```

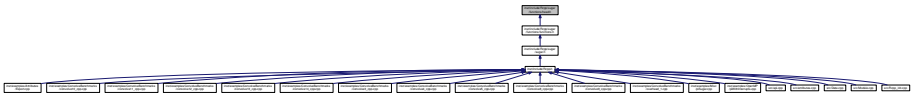
```
#include <Rcpp/sugar/functions/Lazy.h>
#include <Rcpp/sugar/functions/math.h>
#include <Rcpp/sugar/functions/complex.h>
#include <Rcpp/sugar/functions/any.h>
#include <Rcpp/sugar/functions/all.h>
#include <Rcpp/sugar/functions/is_na.h>
#include <Rcpp/sugar/functions/is_finite.h>
#include <Rcpp/sugar/functions/is_infinite.h>
#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>
```

This graph shows which files directly or indirectly include this file:



7.238 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

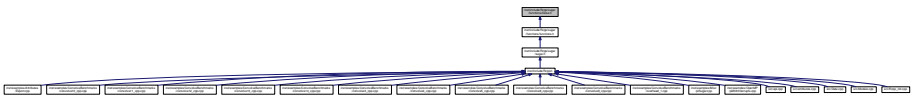
  Rcpp API.
- Rcpp::sugar

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)

7.239 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

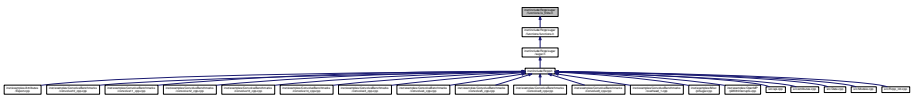
  Rcpp API.
- Rcpp::sugar

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)

7.240 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

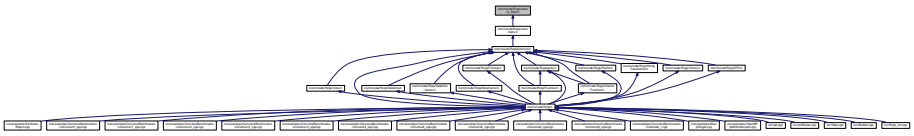
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

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

7.241 inst/include/Rcpp/traits/is_finite.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

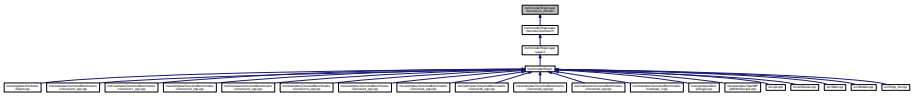
- Rcpp
  Rcpp API.
- Rcpp::traits
  traits used to dispatch wrap

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)

7.242 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)

7.243 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

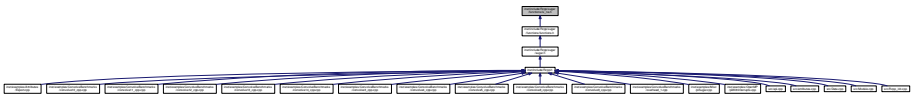
  traits used to dispatch wrap

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)

7.244 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< DatetimeVector > Rcpp::is_na (const DatetimeVector &x)
- sugar::IsNa_Vector_is_na< DateVector > Rcpp::is_na (const DateVector &x)

7.245 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

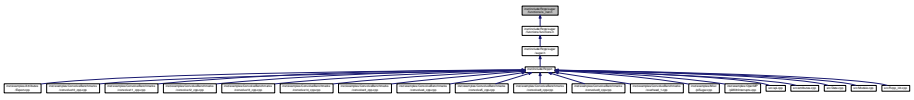
  traits used to dispatch wrap

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)

7.246 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

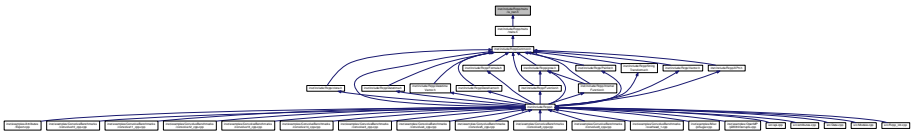
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::IsNaN< RTYPE, NA, T > Rcpp::is_nan (const Rcpp::VectorBase< RTYPE, NA, T > &t)

7.247 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

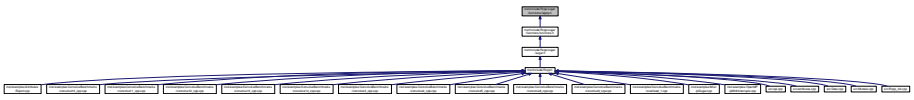
  traits used to dispatch wrap

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)

7.248 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

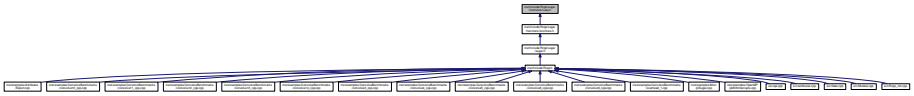
  Rcpp API.
- Rcpp::sugar

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)

7.249 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 >

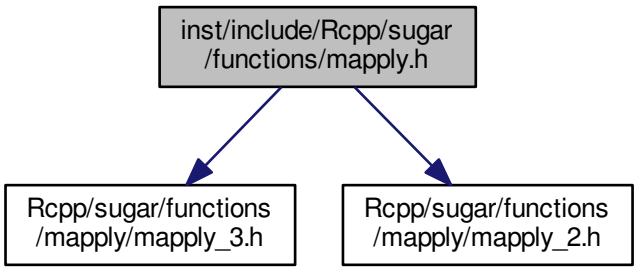
Namespaces

- Rcpp
  Rcpp API.
- Rcpp::sugar

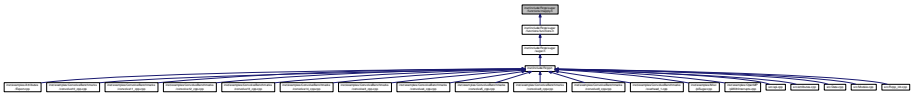
7.250 inst/include/Rcpp/sugar/functions/mapply.h File Reference

```
#include <Rcpp/sugar/functions/mapply/mapply_3.h>
#include <Rcpp/sugar/functions/mapply/mapply_2.h>
```

Include dependency graph for mapply.h:

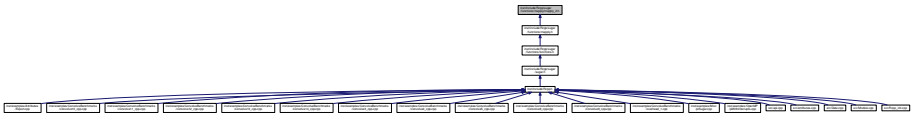


This graph shows which files directly or indirectly include this file:



7.251 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

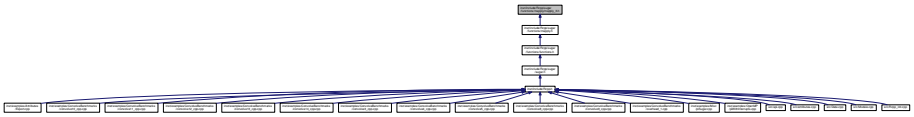
  Rcpp API.
- Rcpp::sugar

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)

7.252 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

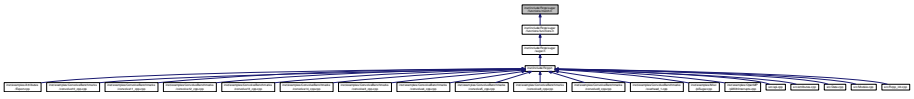
  Rcpp API.
- Rcpp::sugar

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)

7.253 inst/include/Rcpp/sugar/functions/match.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

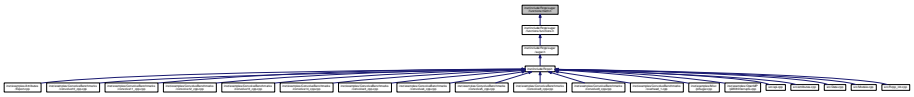
  Rcpp API.

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_)

7.254 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

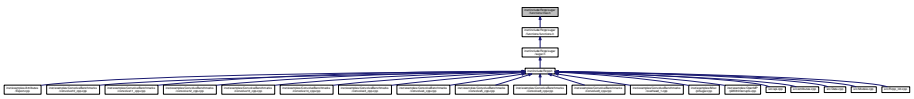
7.254.1 Function Documentation

7.254.1.1 VECTORIZED_MATH_1 (trunc , ::Rf_ftrunc)

Referenced by Rcpp::internal::lfactorial().

7.255 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

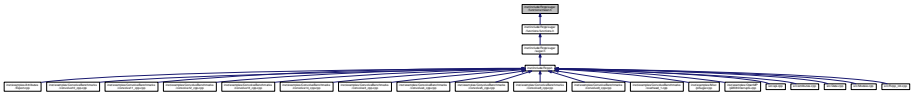
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Max< RTYPE, NA, T > Rcpp::max (const VectorBase< RTYPE, NA, T > &x)

7.256 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

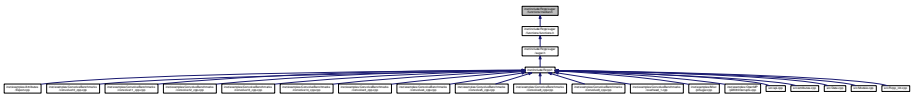
  Rcpp API.
- Rcpp::sugar

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)

7.257 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

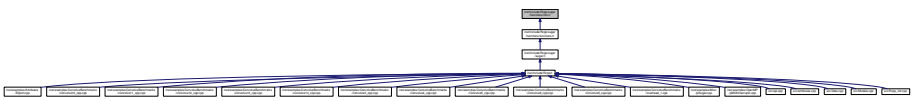
  Rcpp API.
- Rcpp::sugar
- Rcpp::sugar::median_detail

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)

7.258 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

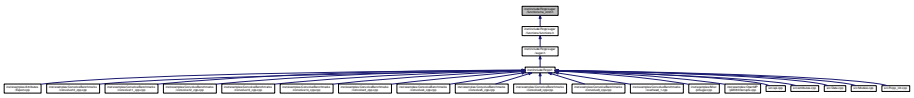
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Min< RTYPE, NA, T > Rcpp::min (const VectorBase< RTYPE, NA, T > &x)

7.259 inst/include/Rcpp/sugar/functions/na_omit.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

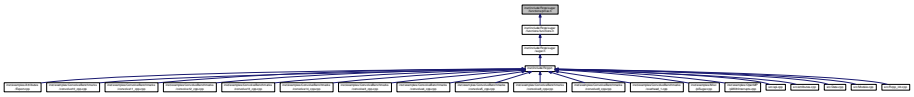
  Rcpp API.
- Rcpp::sugar

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)

7.260 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< RTYPE, LHS_NA, RHS_NA >
- 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

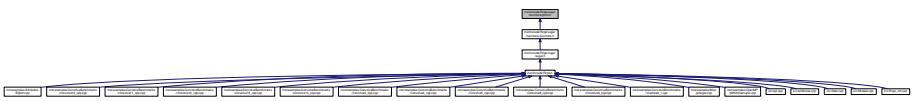
  Rcpp API.
- Rcpp::sugar

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::Vector↩Base< 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)

7.261 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< RTYPE, LHS_NA, RHS_NA >
- 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

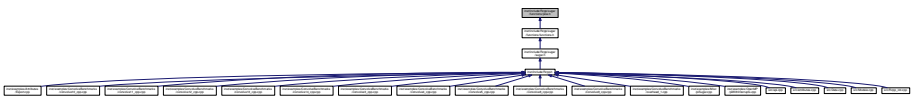
  Rcpp API.
- Rcpp::sugar

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::Vector↩ Base< 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, L↩ HS_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)

7.262 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

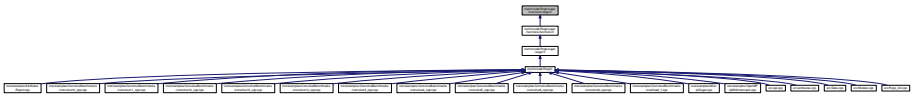
- Rcpp
  Rcpp API.
- Rcpp::sugar

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, EXP↩
  ONENT_TYPE exponent)

7.263 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

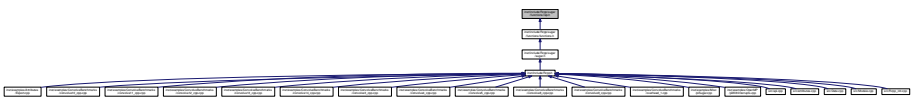
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Range< RTYPE, NA, T > Rcpp::range (const VectorBase< RTYPE, NA, T > &x)

7.264 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

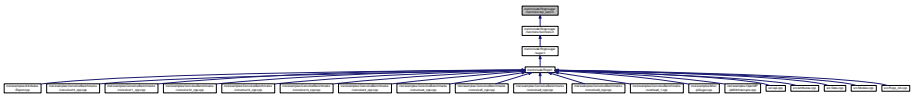
  Rcpp API.
- Rcpp::sugar

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)

7.265 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

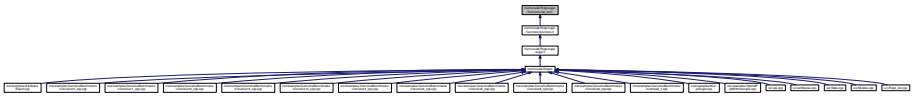
  Rcpp API.
- Rcpp::sugar

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)

7.266 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

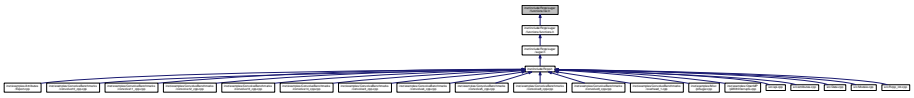
  Rcpp API.
- Rcpp::sugar

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)

7.267 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

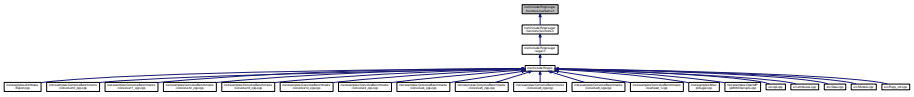
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Rev< RTYPE, NA, T > Rcpp::rev (const VectorBase< RTYPE, NA, T > &t)

7.268 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)

7.268.1 Macro Definition Documentation

7.268.1.1 #define COL_MEANS_IMPL_KEEPNA(__RTYPE__)

Definition at line 753 of file rowSums.h.

7.268.1.2 #define COL_MEANS_IMPL_RMNA(__RTYPE__)

Definition at line 861 of file rowSums.h.

7.268.1.3 #define COL_SUMS_IMPL_KEEPNA(__RTYPE__)

Definition at line 353 of file rowSums.h.

7.268.1.4 #define COL_SUMS_IMPL_RMNA(__RTYPE__)

Definition at line 448 of file rowSums.h.

7.268.1.5 #define ROW_MEANS_IMPL_KEEPNA(__RTYPE__)

Definition at line 544 of file rowSums.h.

7.268.1.6 #define ROW_MEANS_IMPL_RMNA(__RTYPE__)

Definition at line 652 of file rowSums.h.

7.268.1.7 #define ROW_SUMS_IMPL_KEEPNA(__RTYPE__)

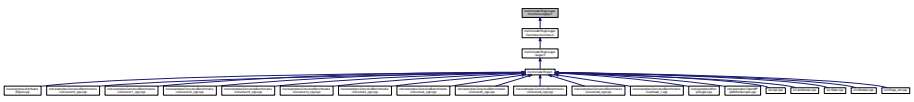
Definition at line 169 of file rowSums.h.

7.268.1.8 #define ROW_SUMS_IMPL_RMNA(__RTYPE__)

Definition at line 264 of file rowSums.h.

7.269 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

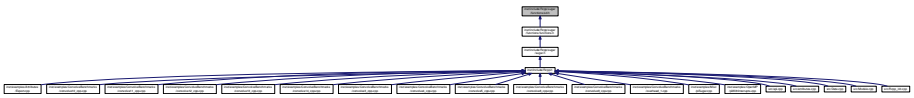
- Rcpp
  Rcpp API.
- Rcpp::sugar

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_result_of< Function, T >::type >::rtype >::type >::value > Rcpp::sapply (const Rcpp::VectorBase< RTYPE, NA, T > &t, Function fun)

7.270 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

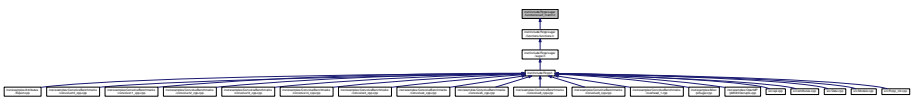
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Sd< REALSXP, NA, T > Rcpp::sd (const VectorBase< REALSXP, NA, T > &t)

7.271 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

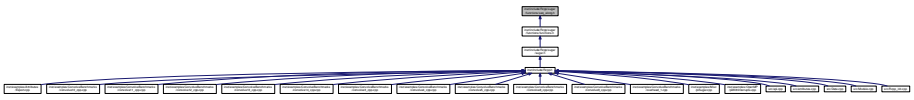
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  IntegerVector Rcpp::self_match (const VectorBase< RTYPE, NA, T > &x)

7.272 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

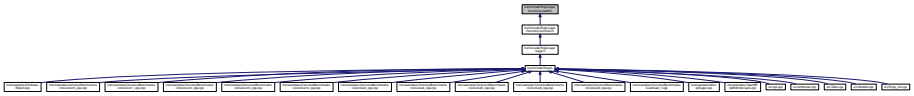
  Rcpp API.
- Rcpp::sugar

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)

7.273 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

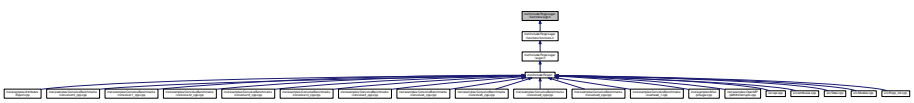
  Rcpp API.
- Rcpp::sugar

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< R↩
  TYPE, 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)

7.274 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

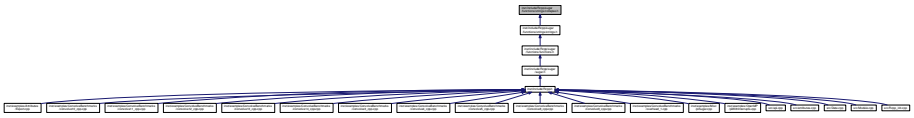
  Rcpp API.
- Rcpp::sugar

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)

7.275 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

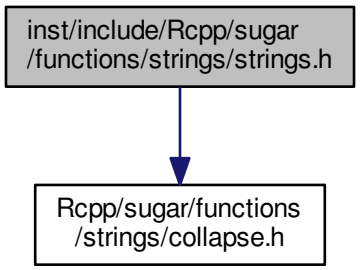
Functions

- 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)

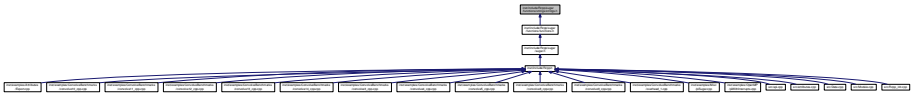
7.276 inst/include/Rcpp/sugar/functions/strings/strings.h File Reference

```
#include <Rcpp/sugar/functions/strings/collapse.h>
```

Include dependency graph for strings.h:

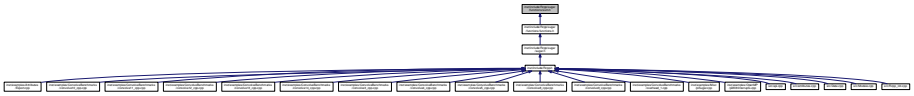


This graph shows which files directly or indirectly include this file:



7.277 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

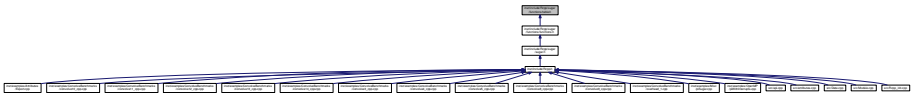
  Rcpp API.
- Rcpp::sugar

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)

7.278 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

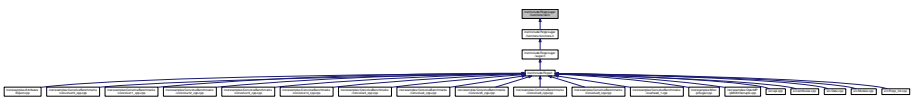
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  IntegerVector Rcpp::table (const VectorBase< RTYPE, NA, T > &x)

7.279 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

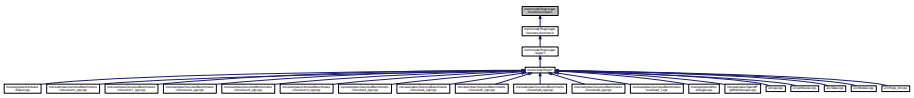
  Rcpp API.
- Rcpp::sugar

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)

7.280 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

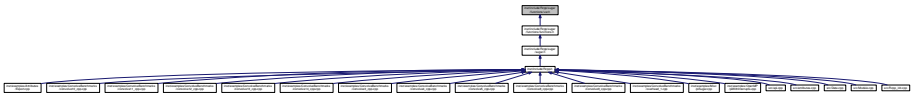
  Rcpp API.
- Rcpp::sugar

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)
- 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)

7.281 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

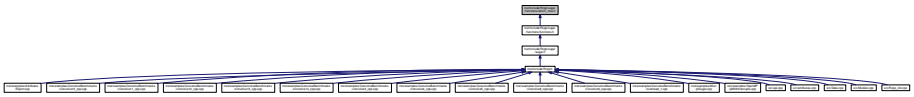
  Rcpp API.
- Rcpp::sugar

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)

7.282 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

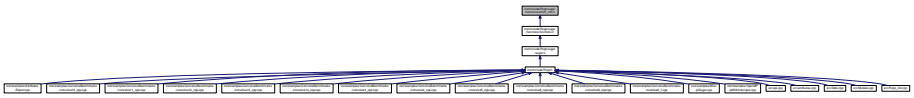
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  R_xlen_t Rcpp::which_max (const VectorBase< RTYPE, NA, T > &t)

7.283 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

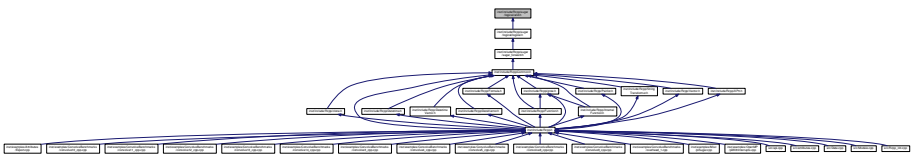
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  R_xlen_t Rcpp::which_min (const VectorBase< RTYPE, NA, T > &t)

7.284 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)

7.284.1 Function Documentation

7.284.1.1 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.

References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.284.1.2 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.

References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.284.1.3 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.

References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

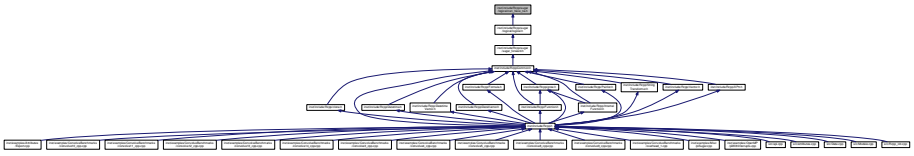
7.284.1.4 template<bool LHS_NA, typename LHS_T > Rcpp::sugar::And_SingleLogicalResult_bool<LHS_NA,LHS_T> operator&& (bool *rhs,* const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & *lhs*) [inline]

Definition at line 306 of file and.h.

References Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.285 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

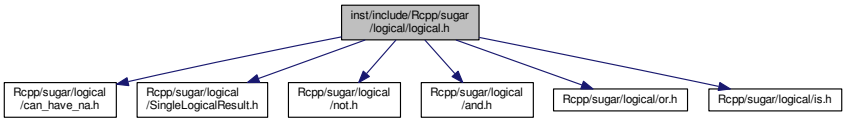
- Rcpp

  Rcpp API.

7.286 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>
```

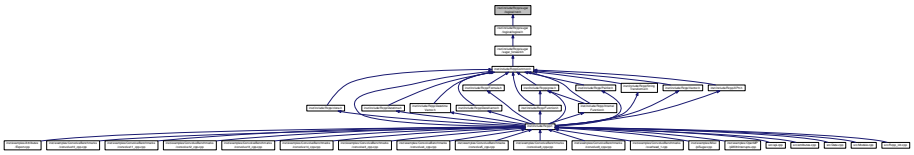
Include dependency graph for logical.h:



This graph shows which files directly or indirectly include this file:

7.287 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)

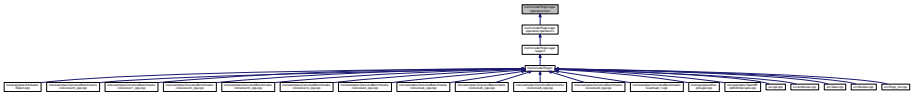
7.287.1 Function Documentation

7.287.1.1 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.

7.288 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)

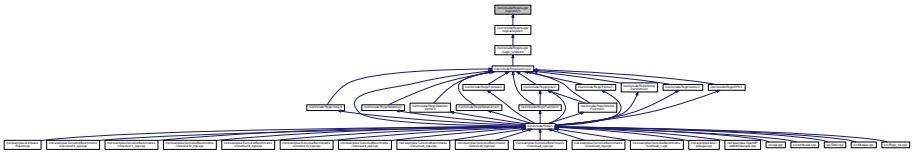
7.288.1 Function Documentation

7.288.1.1 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.

7.289 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::SingleLogical↩ Result< 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)

7.289.1 Function Documentation

7.289.1.1 template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T > Rcpp::sugar::Or_Logical↩ Expression_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.

References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.289.1.2 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.

References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.289.1.3 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.

References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

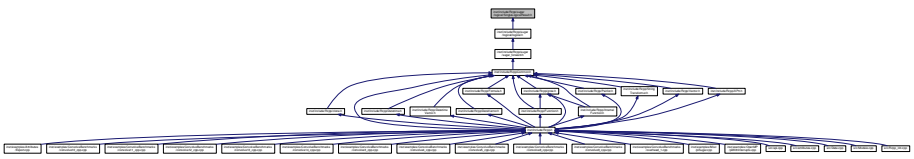
7.289.1.4 template<bool LHS_NA, typename LHS_T > Rcpp::sugar::Or_SingleLogicalResult_bool<LHS_NA,LHS_T> operator|| (bool *rhs,* const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & *lhs*) [inline]

Definition at line 305 of file or.h.

References Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

7.290 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< bool >
- class Rcpp::sugar::forbidden_conversion< true >
- class Rcpp::sugar::conversion_to_bool_is_forbidden< x >
- class Rcpp::sugar::SingleLogicalResult< NA, T >

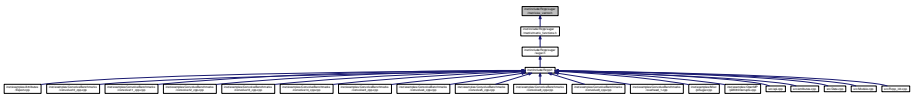
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

7.291 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

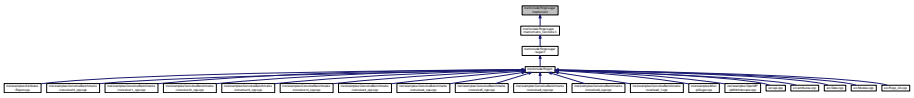
  internal implementation details

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)

7.292 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

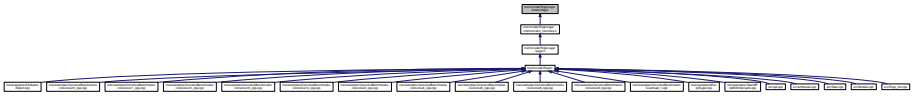
- Rcpp
  Rcpp API.
- Rcpp::sugar

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)

7.293 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

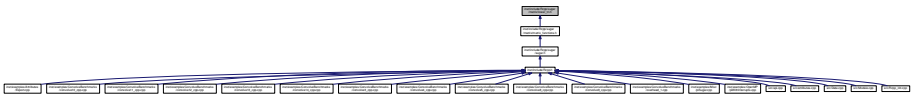
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<typename T >
  sugar::diag_result_type_trait< T >::type Rcpp::diag (const T &t)

7.294 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, LHS_NA, LHS_T >

Namespaces

- Rcpp
  Rcpp API.
- Rcpp::sugar

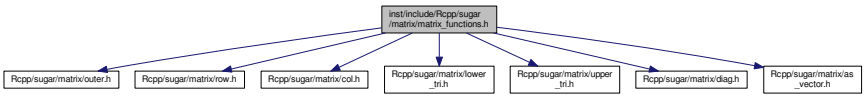
Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::LowerTri< RTYPE, LHS_NA, LHS_T > Rcpp::lower_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, L↩
  HS_T > &lhs, bool diag=false)

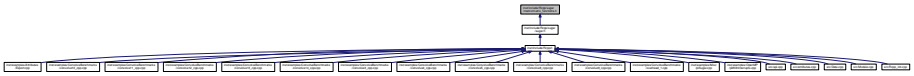
7.295 inst/include/Rcpp/sugar/matrix/matrix_functions.h File Reference

```
#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:

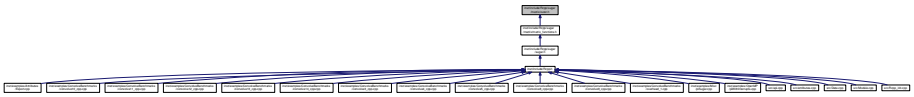


This graph shows which files directly or indirectly include this file:



7.296 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

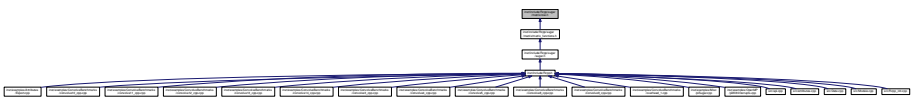
  Rcpp API.
- Rcpp::sugar

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)

7.297 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

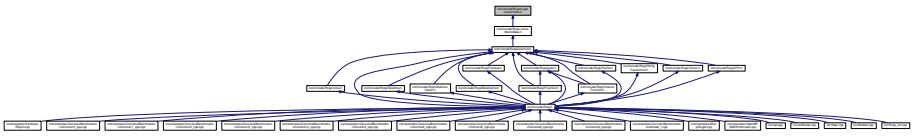
  Rcpp API.
- Rcpp::sugar

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)

7.298 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

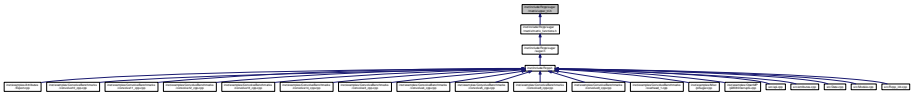
  internal implementation details

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)

7.299 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, LHS_NA, LHS_T >

Namespaces

- Rcpp

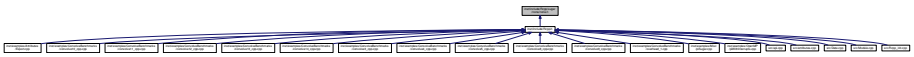
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::UpperTri< RTYPE, LHS_NA, LHS_T > Rcpp::upper_tri (const Rcpp::MatrixBase< RTYPE, LHS_NA, L↩
  HS_T > &lhs, bool diag=false)

7.300 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

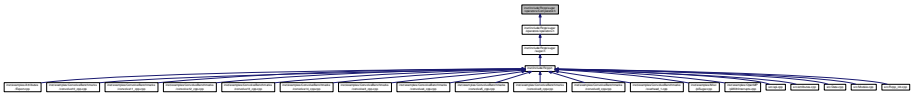
  Rcpp API.
- Rcpp::sugar

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)

7.301 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 >

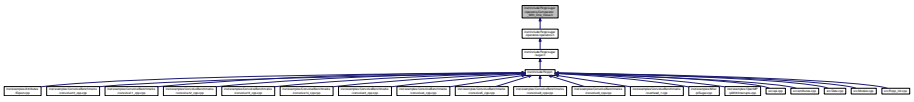
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

7.302 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 >

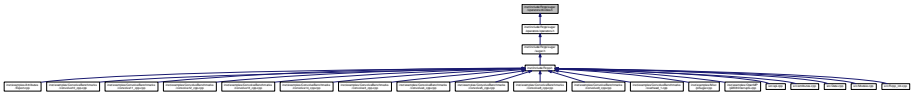
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

7.303 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

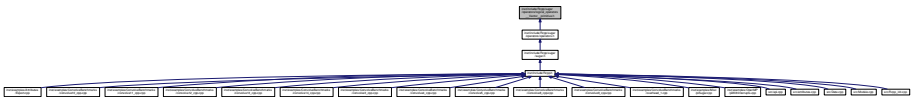
  Rcpp API.
- Rcpp::sugar

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)

7.304 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> (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< 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< RT↩YPE >::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)

7.304.1 Function Documentation

7.304.1.1 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.

7.304.1.2 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.

7.304.1.3 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 98 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.304.1.4 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 34 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.304.1.5 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 122 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.304.1.6 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 187 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.304.1.7 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.

7.304.1.8 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.

7.304.1.9 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.

7.304.1.10 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.

References Rcpp::NA.

7.304.1.11 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.

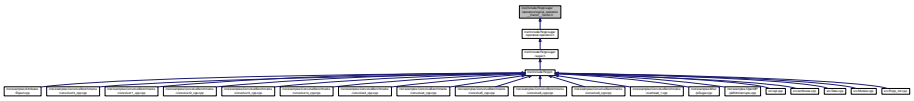
7.304.1.12 template<int RTYPE, bool NA, typename T > Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, NA, T > operator>= (const Rcpp::VectorBase< RTYPE, NA, T > & *lhs,* typename Rcpp::traits::storage_type< RTYPE >::type *rhs*) [inline]

Definition at line 167 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.305 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, R↩ HS_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, R↩ HS_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)

7.305.1 Function Documentation

7.305.1.1 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.

7.305.1.2 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 34 of file logical_operators__Vector__Vector.h.

7.305.1.3 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 76 of file logical_operators__Vector__Vector.h.

7.305.1.4 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.

7.305.1.5 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]

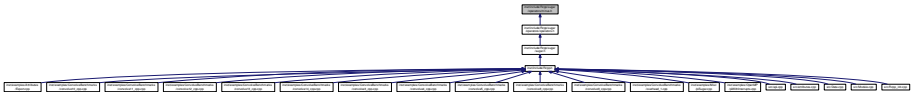
Definition at line 55 of file logical_operators__Vector__Vector.h.

7.305.1.6 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.

7.306 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

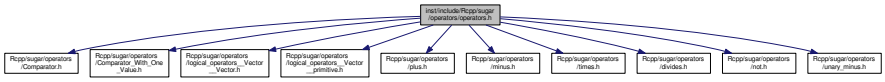
Functions

- 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 Vector↩ Base< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

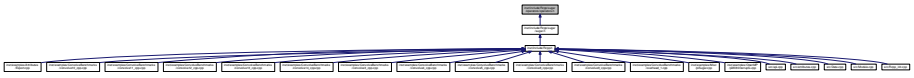
7.307 inst/include/Rcpp/sugar/operators/operators.h File Reference

```
#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:

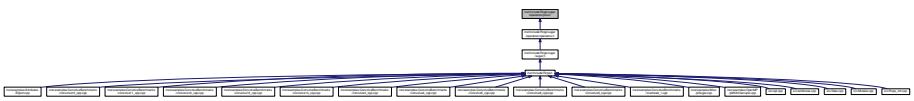


This graph shows which files directly or indirectly include this file:



7.308 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)

7.309 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)

7.309.1 Macro Definition Documentation

7.309.1.1 #define RCPP_OP(*NAME, OP*)

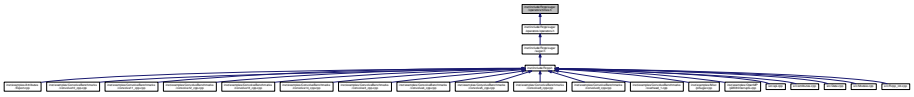
Value:

```
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.

7.310 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

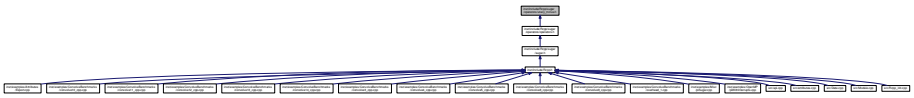
  Rcpp API.
- Rcpp::sugar

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 Vector↩Base< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.311 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)

7.311.1 Function Documentation

7.311.1.1 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.

7.312 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.

7.313 inst/include/Rcpp/sugar/sets.h File Reference

Macros

- #define RCPP_UNORDERED_SET std::set
- #define RCPP_UNORDERED_MAP std::map

7.313.1 Macro Definition Documentation

7.313.1.1 #define RCPP_UNORDERED_MAP std::map

Definition at line 58 of file sets.h.

7.313.1.2 #define RCPP_UNORDERED_SET std::set

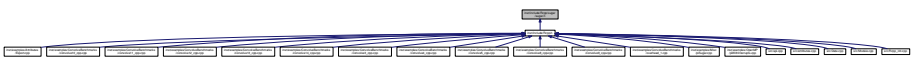
Definition at line 57 of file sets.h.

7.314 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:

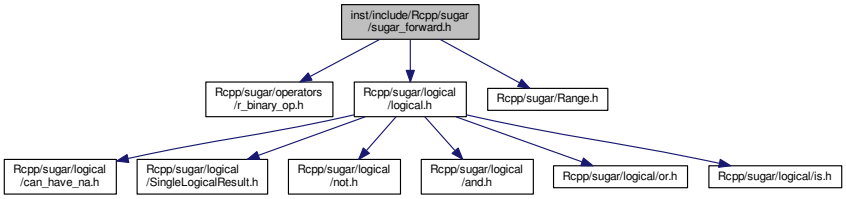
This graph shows which files directly or indirectly include this file:



7.315 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>
```

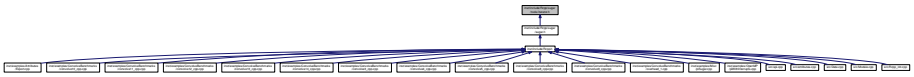
Include dependency graph for sugar_forward.h:



This graph shows which files directly or indirectly include this file:

7.316 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

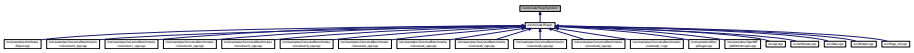
- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_begin__impl (const T &obj, Rcpp::traits::true_type)
- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_begin__impl (const T &obj, Rcpp::traits::false_type)
- template<typename T >
  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)

7.317 inst/include/Rcpp/sugar/undoRmath.h File Reference

This graph shows which files directly or indirectly include this file:

7.318 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)

7.319 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

7.320 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

7.321 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< LGLSXP >
- struct Rcpp::traits::expands_to_logical__impl< LGLSXP >::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

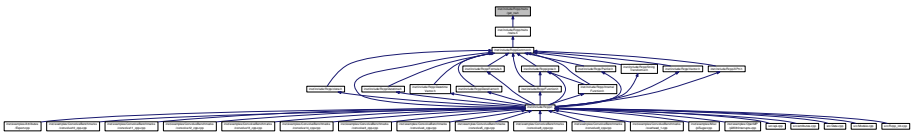
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.322 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

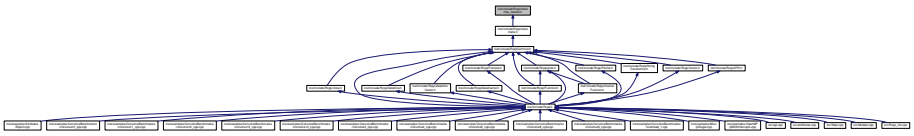
  traits used to dispatch wrap

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 > ()

7.323 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

7.324 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

7.325 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

7.326 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

7.327 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

7.328 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

7.329 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

7.330 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

7.331 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

7.332 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

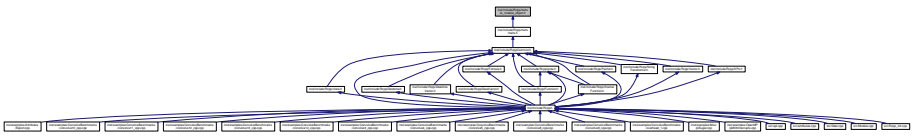
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.333 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

7.334 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

7.335 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

7.336 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

7.337 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

7.338 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

7.339 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

7.340 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

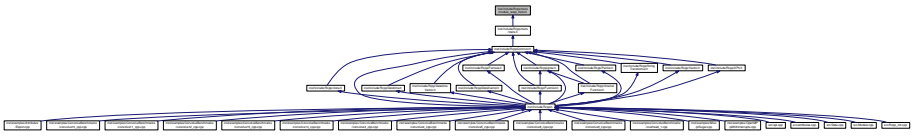
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.341 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

7.342 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

7.343 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

7.344 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_needscast< T >
- struct Rcpp::traits::r_sexptype_needscast< int >
- struct Rcpp::traits::r_sexptype_needscast< double >
- struct Rcpp::traits::r_sexptype_needscast< Rcomplex >
- struct Rcpp::traits::r_sexptype_needscast< Rbyte >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.345 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

7.346 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

7.347 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

7.348 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

7.349 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

7.350 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

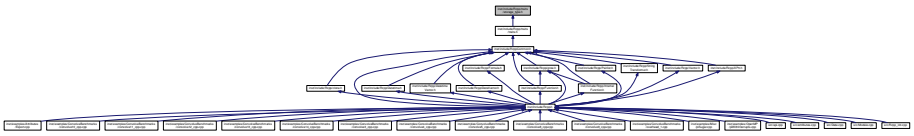
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.351 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

7.352 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

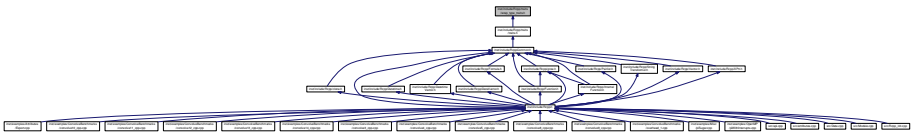
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.353 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

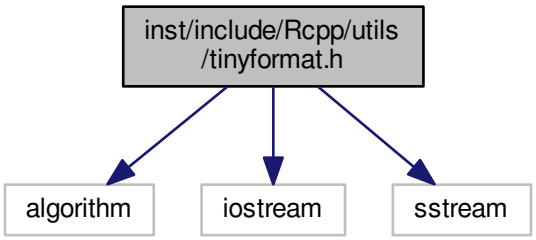
  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.354 inst/include/Rcpp/utils/tinyformat.h File Reference

```
#include <algorithm>
#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
- Rcpp

  Rcpp API.
- tinyformat::detail

Macros

- #define TINYFORMAT_ERROR(REASON) ::Rcpp::stop(REASON)
- #define TINYFORMAT_NO_VARIADIC_TEMPLATES
- #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

- void NORET Rcpp::stop (const std::string &message)
- 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::printfln (const char *fmt)

7.354.1 Macro Definition Documentation

7.354.1.1 #define TINYFORMAT_ARGTYPES(*n*) TINYFORMAT_ARGTYPES_ ## n

Definition at line 380 of file tinyformat.h.

7.354.1.2 #define TINYFORMAT_ARGTYPES_1 class T1

Definition at line 421 of file tinyformat.h.

7.354.1.3 #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 430 of file tinyformat.h.

7.354.1.4 #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 431 of file tinyformat.h.

7.354.1.5 #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 432 of file tinyformat.h.

7.354.1.6 #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 433 of file tinyformat.h.

7.354.1.7 #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 434 of file tinyformat.h.

7.354.1.8 #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 435 of file tinyformat.h.

7.354.1.9 #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 436 of file tinyformat.h.

7.354.1.10 #define TINYFORMAT_ARGTYPES_2 class T1, class T2

Definition at line 422 of file tinyformat.h.

7.354.1.11 #define TINYFORMAT_ARGTYPES_3 class T1, class T2, class T3

Definition at line 423 of file tinyformat.h.

7.354.1.12 #define TINYFORMAT_ARGTYPES_4 class T1, class T2, class T3, class T4

Definition at line 424 of file tinyformat.h.

7.354.1.13 #define TINYFORMAT_ARGTYPES_5 class T1, class T2, class T3, class T4, class T5

Definition at line 425 of file tinyformat.h.

7.354.1.14 #define TINYFORMAT_ARGTYPES_6 class T1, class T2, class T3, class T4, class T5, class T6

Definition at line 426 of file tinyformat.h.

7.354.1.15 #define TINYFORMAT_ARGTYPES_7 class T1, class T2, class T3, class T4, class T5, class T6, class T7

Definition at line 427 of file tinyformat.h.

7.354.1.16 #define TINYFORMAT_ARGTYPES_8 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8

Definition at line 428 of file tinyformat.h.

7.354.1.17 #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 429 of file tinyformat.h.

7.354.1.18 #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 287 of file tinyformat.h.

7.354.1.19 #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 356 of file tinyformat.h.

7.354.1.20 #define TINYFORMAT_ERROR(*REASON*) ::Rcpp::stop(REASON)

Definition at line 129 of file tinyformat.h.

Referenced by tinyformat::detail::formatImpl(), tinyformat::detail::convertToInt< T, convertible >::invoke(), and tinyformat::detail::streamStateFromFormat().

7.354.1.21 #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 489 of file tinyformat.h.

7.354.1.22 #define TINYFORMAT_HIDDEN

Definition at line 175 of file tinyformat.h.

7.354.1.23 #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 printfln(const char* fmt, TINYFORMAT_VARARGS(n))                   \
{                                                                       \
    format(std::cout, fmt, TINYFORMAT_PASSARGS(n));                     \
    std::cout << '\n';                                                  \
}
```

Definition at line 1014 of file tinyformat.h.

7.354.1.24 #define TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR(*n*)

Value:

```
\
        template<TINYFORMAT_ARGTYPES(n)>                        \
        FormatListN(TINYFORMAT_VARARGS(n))                      \
            : FormatList(&m_formatterStore[0], n)               \
        {/*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 876 of file tinyformat.h.

7.354.1.25 #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 934 of file tinyformat.h.

7.354.1.26 #define TINYFORMAT_NO_VARIADIC_TEMPLATES

Definition at line 136 of file tinyformat.h.

7.354.1.27 #define TINYFORMAT_PASSARGS(*n*) TINYFORMAT_PASSARGS_ ## n

Definition at line 382 of file tinyformat.h.

7.354.1.28 #define TINYFORMAT_PASSARGS_1 v1

Definition at line 455 of file tinyformat.h.

7.354.1.29 #define TINYFORMAT_PASSARGS_10 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

Definition at line 464 of file tinyformat.h.

7.354.1.30 #define TINYFORMAT_PASSARGS_11 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11

Definition at line 465 of file tinyformat.h.

7.354.1.31 #define TINYFORMAT_PASSARGS_12 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12

Definition at line 466 of file tinyformat.h.

7.354.1.32 #define TINYFORMAT_PASSARGS_13 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13

Definition at line 467 of file tinyformat.h.

7.354.1.33 #define TINYFORMAT_PASSARGS_14 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14

Definition at line 468 of file tinyformat.h.

7.354.1.34 #define TINYFORMAT_PASSARGS_15 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15

Definition at line 469 of file tinyformat.h.

7.354.1.35 #define TINYFORMAT_PASSARGS_16 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16

Definition at line 470 of file tinyformat.h.

7.354.1.36 #define TINYFORMAT_PASSARGS_2 v1, v2

Definition at line 456 of file tinyformat.h.

7.354.1.37 #define TINYFORMAT_PASSARGS_3 v1, v2, v3

Definition at line 457 of file tinyformat.h.

7.354.1.38 #define TINYFORMAT_PASSARGS_4 v1, v2, v3, v4

Definition at line 458 of file tinyformat.h.

7.354.1.39 #define TINYFORMAT_PASSARGS_5 v1, v2, v3, v4, v5

Definition at line 459 of file tinyformat.h.

7.354.1.40 #define TINYFORMAT_PASSARGS_6 v1, v2, v3, v4, v5, v6

Definition at line 460 of file tinyformat.h.

7.354.1.41 #define TINYFORMAT_PASSARGS_7 v1, v2, v3, v4, v5, v6, v7

Definition at line 461 of file tinyformat.h.

7.354.1.42 #define TINYFORMAT_PASSARGS_8 v1, v2, v3, v4, v5, v6, v7, v8

Definition at line 462 of file tinyformat.h.

7.354.1.43 #define TINYFORMAT_PASSARGS_9 v1, v2, v3, v4, v5, v6, v7, v8, v9

Definition at line 463 of file tinyformat.h.

7.354.1.44 #define TINYFORMAT_PASSARGS_TAIL(*n*) TINYFORMAT_PASSARGS_TAIL_ ## n

Definition at line 383 of file tinyformat.h.

7.354.1.45 #define TINYFORMAT_PASSARGS_TAIL_1

Definition at line 472 of file tinyformat.h.

7.354.1.46 #define TINYFORMAT_PASSARGS_TAIL_10 , v2, v3, v4, v5, v6, v7, v8, v9, v10

Definition at line 481 of file tinyformat.h.

7.354.1.47 #define TINYFORMAT_PASSARGS_TAIL_11 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11

Definition at line 482 of file tinyformat.h.

7.354.1.48 #define TINYFORMAT_PASSARGS_TAIL_12 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12

Definition at line 483 of file tinyformat.h.

7.354.1.49 #define TINYFORMAT_PASSARGS_TAIL_13 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13

Definition at line 484 of file tinyformat.h.

7.354.1.50 #define TINYFORMAT_PASSARGS_TAIL_14 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14

Definition at line 485 of file tinyformat.h.

7.354.1.51 #define TINYFORMAT_PASSARGS_TAIL_15 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15

Definition at line 486 of file tinyformat.h.

7.354.1.52 #define TINYFORMAT_PASSARGS_TAIL_16 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16

Definition at line 487 of file tinyformat.h.

7.354.1.53 #define TINYFORMAT_PASSARGS_TAIL_2 , v2

Definition at line 473 of file tinyformat.h.

7.354.1.54 #define TINYFORMAT_PASSARGS_TAIL_3 , v2, v3

Definition at line 474 of file tinyformat.h.

7.354.1.55 #define TINYFORMAT_PASSARGS_TAIL_4 , v2, v3, v4

Definition at line 475 of file tinyformat.h.

7.354.1.56 #define TINYFORMAT_PASSARGS_TAIL_5 , v2, v3, v4, v5

Definition at line 476 of file tinyformat.h.

7.354.1.57 #define TINYFORMAT_PASSARGS_TAIL_6 , v2, v3, v4, v5, v6

Definition at line 477 of file tinyformat.h.

7.354.1.58 #define TINYFORMAT_PASSARGS_TAIL_7 , v2, v3, v4, v5, v6, v7

Definition at line 478 of file tinyformat.h.

7.354.1.59 #define TINYFORMAT_PASSARGS_TAIL_8 , v2, v3, v4, v5, v6, v7, v8

Definition at line 479 of file tinyformat.h.

7.354.1.60 #define TINYFORMAT_PASSARGS_TAIL_9 , v2, v3, v4, v5, v6, v7, v8, v9

Definition at line 480 of file tinyformat.h.

7.354.1.61 #define TINYFORMAT_VARARGS(*n*) TINYFORMAT_VARARGS_ ## n

Definition at line 381 of file tinyformat.h.

7.354.1.62 #define TINYFORMAT_VARARGS_1 const T1& v1

Definition at line 438 of file tinyformat.h.

7.354.1.63 #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 447 of file tinyformat.h.

7.354.1.64 #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 448 of file tinyformat.h.

7.354.1.65 #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 449 of file tinyformat.h.

7.354.1.66 #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 450 of file tinyformat.h.

7.354.1.67 #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 451 of file tinyformat.h.

7.354.1.68 #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 452 of file tinyformat.h.

7.354.1.69 #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 453 of file tinyformat.h.

7.354.1.70 #define TINYFORMAT_VARARGS_2 const T1& v1, const T2& v2

Definition at line 439 of file tinyformat.h.

7.354.1.71 #define TINYFORMAT_VARARGS_3 const T1& v1, const T2& v2, const T3& v3

Definition at line 440 of file tinyformat.h.

7.354.1.72 #define TINYFORMAT_VARARGS_4 const T1& v1, const T2& v2, const T3& v3, const T4& v4

Definition at line 441 of file tinyformat.h.

7.354.1.73 #define TINYFORMAT_VARARGS_5 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5

Definition at line 442 of file tinyformat.h.

7.354.1.74 #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 443 of file tinyformat.h.

7.354.1.75 #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 444 of file tinyformat.h.

7.354.1.76 #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

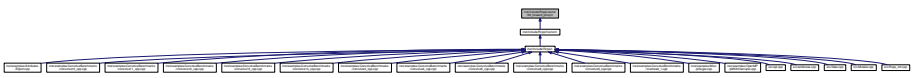
Definition at line 445 of file tinyformat.h.

7.354.1.77 #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 446 of file tinyformat.h.

7.355 inst/include/Rcpp/vector/00_forward_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::string_proxy< RTYPE >
- class Rcpp::internal::const_string_proxy< RTYPE >
- class Rcpp::internal::generic_proxy< RTYPE >
- class Rcpp::internal::const_generic_proxy< RTYPE >
- class Rcpp::internal::simple_name_proxy< RTYPE >
- class Rcpp::internal::string_name_proxy< RTYPE >
- class Rcpp::internal::generic_name_proxy< RTYPE >
- struct Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >
- class Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_name_proxy< RTYPE >
- struct Rcpp::traits::r_vector_proxy< RTYPE >
- struct Rcpp::traits::r_vector_const_proxy< RTYPE >
- struct Rcpp::traits::r_vector_iterator< RTYPE >
- struct Rcpp::traits::r_vector_const_iterator< RTYPE >

Namespaces

- Rcpp

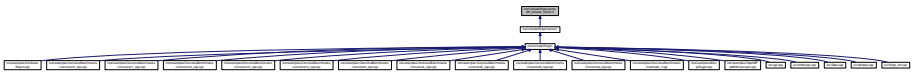
  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

7.356 inst/include/Rcpp/vector/00_forward_Vector.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Vector< RTYPE, StoragePolicy >

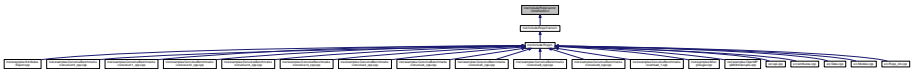
Namespaces

- Rcpp

  Rcpp API.

7.357 inst/include/Rcpp/vector/ChildVector.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

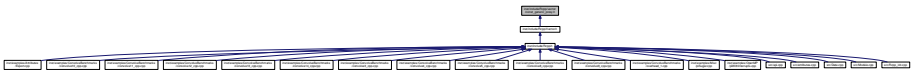
- class Rcpp::ChildVector< T >

Namespaces

- Rcpp
  Rcpp API.

7.358 inst/include/Rcpp/vector/const_generic_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

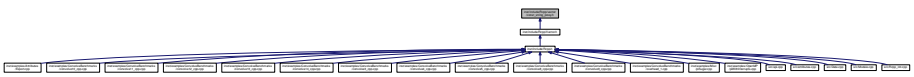
- class Rcpp::internal::const_generic_proxy< RTYPE >

Namespaces

- Rcpp
  Rcpp API.
- Rcpp::internal
  internal implementation details

7.359 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 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

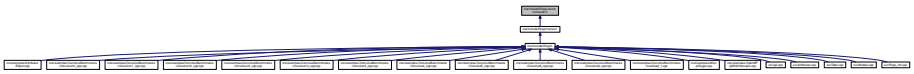
  internal implementation details

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)
- std::ostream & Rcpp::internal::operator<< (std::ostream &os, const const_string_proxy< STRSXP > &proxy)
- std::string Rcpp::internal::operator+ (const std::string &x, const const_string_proxy< STRSXP > &y)

7.360 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

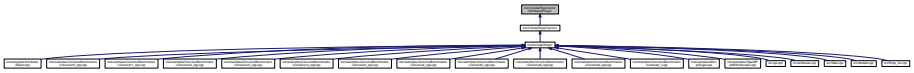
  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

7.361 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

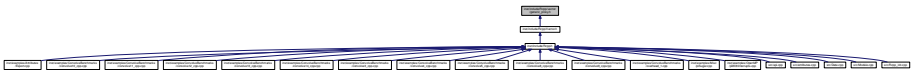
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.362 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 >

Namespaces

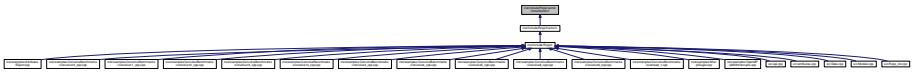
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.363 inst/include/Rcpp/vector/instantiation.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

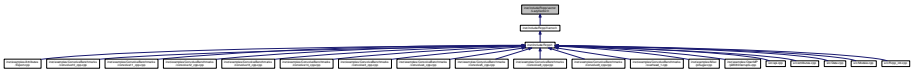
  Rcpp API.

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

7.364 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

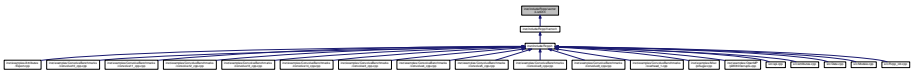
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.365 inst/include/Rcpp/vector/ListOf.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::ListOf< T >
- class Rcpp::sugar::Lapply< RTYPE, NA, T, Function >
- class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >

Namespaces

- Rcpp

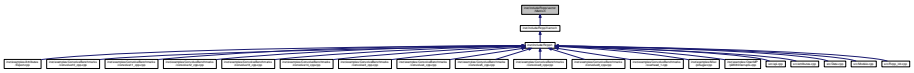
  Rcpp API.
- Rcpp::sugar

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)

7.366 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::tranpose_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)

7.366.1 Macro Definition Documentation

7.366.1.1 #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.

7.366.1.2 #define RCPP_GENERATE_SCALAR_MATRIX_OPERATOR(__OPERATOR__)

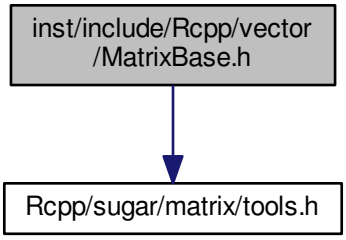
Value:

```
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 = static_cast<const Vector<RTYPE, StoragePolicy> &>(rhs);
                         \
         v = lhs __OPERATOR__ v;
                         \
         v.attr("dim") = Vector<INTSXP>::create(rhs.nrow(), rhs.ncol());
                                                \
         return as< Matrix<RTYPE, StoragePolicy> >(v);
                         \
    }
```

Definition at line 267 of file Matrix.h.

7.367 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

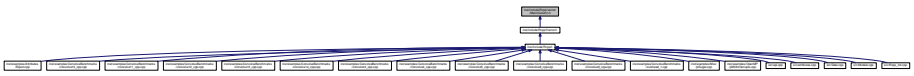
Namespaces

- Rcpp

  Rcpp API.

7.368 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 >

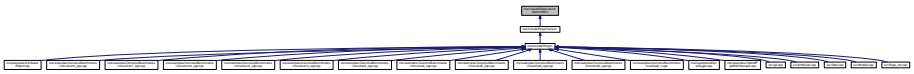
Namespaces

- Rcpp

  Rcpp API.

7.369 inst/include/Rcpp/vector/MatrixRow.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::MatrixRow< RTYPE >
- class Rcpp::MatrixRow< RTYPE >::iterator
- class Rcpp::ConstMatrixRow< RTYPE >
- class Rcpp::ConstMatrixRow< RTYPE >::const_iterator

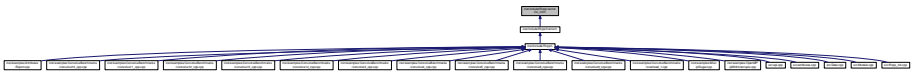
Namespaces

- Rcpp

  Rcpp API.

7.370 inst/include/Rcpp/vector/no_init.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Matrix< RTYPE, StoragePolicy >
- class Rcpp::no_init_vector
- class Rcpp::no_init_matrix

Namespaces

- Rcpp

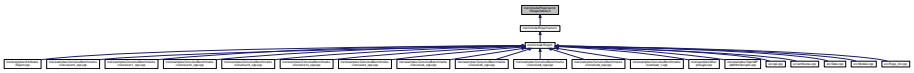
  Rcpp API.

Functions

- no_init_vector Rcpp::no_init (int size)
- no_init_matrix Rcpp::no_init (int nr, int nc)

7.371 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)

7.371.1 Macro Definition Documentation

7.371.1.1 #define UNROLL_LOOP(*OP*)

Value:

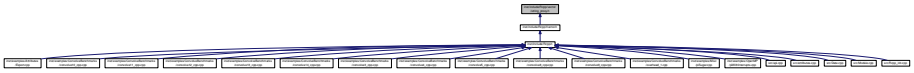
```
typedef typename ::Rcpp::traits::Extractor<RTYPE,NA,T>::type EXT ; \
    const EXT& input( x.get_ref() ) ;                              \
    int __trip_count = (size_) >> 2;                              \
    int i=0 ;                                                     \
    for ( ; __trip_count > 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.

Referenced by Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator*=(), Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator+=(), Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator-=(), Rcpp↩ ::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator/=(), and Rcpp::internal::RangeIndexer< RTYPE, NA, V↩ ECTOR >::operator=().

7.372 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 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

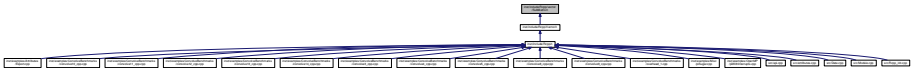
  internal implementation details

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)

7.373 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)

7.373.1 Macro Definition Documentation

7.373.1.1 #define RCPP_WRAP_SUBMATRIX(*RTYPE*)

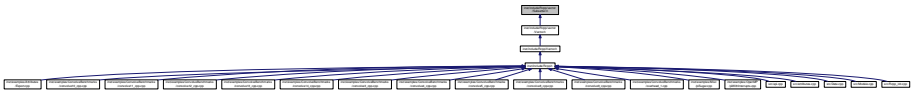
Value:

```
template<> inline SEXP wrap< SubMatrix<RTYPE> >(  \
    const SubMatrix<RTYPE>& object                \
    ) {                                           \
        return Matrix<RTYPE>( object ) ;          \
    }
```

Definition at line 90 of file SubMatrix.h.

7.374 inst/include/Rcpp/vector/Subsetter.h File Reference

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__)

7.374.1 Macro Definition Documentation

7.374.1.1 #define RCPP_GENERATE_SUBSET_PROXY_OPERATOR(*__OPERATOR__*)

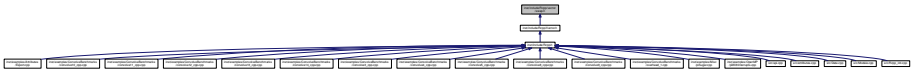
Value:

```
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 (int 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 (int 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 229 of file Subsetter.h.

7.375 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)

7.375.1 Macro Definition Documentation

7.375.1.1 #define RCPP_GENERATE_SWAP(*TYPE, RTYPE*)

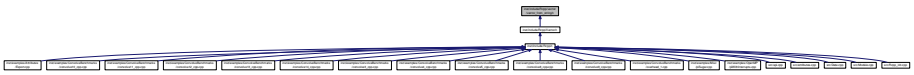
Value:

```
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.

7.376 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< EXPRSXP > (const std::string &st)

7.377 inst/include/Rcpp/vector/VectorBase.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

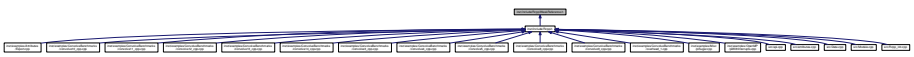
- class Rcpp::VectorBase< RTYPE, na, VECTOR >
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na
- class Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator

Namespaces

- Rcpp
  Rcpp API.

7.378 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

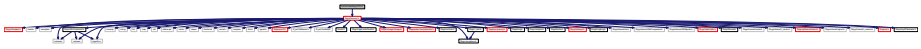
Functions

- Rcpp::RCPP_API_CLASS (WeakReference_Impl)

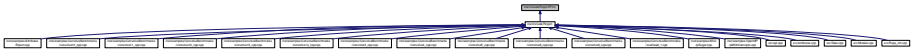
7.379 inst/include/Rcpp/XPtr.h File Reference

```
#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 >

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)

7.380 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 <limits>
#include <typeinfo>
#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:

Classes

- class Rcpp::traits::named_object< T >
- class Rcpp::object< T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap
- Rcpp::internal

  internal implementation details

Macros

- #define RcppExport extern "C"

Functions

- SEXP Rcpp::Rcpp_eval (SEXP expr, SEXP env)
- SEXP Rcpp::Rcpp_PreserveObject (SEXP x)
- void Rcpp::Rcpp_ReleaseObject (SEXP x)
- SEXP Rcpp::Rcpp_ReplaceObject (SEXP x, SEXP y)
- template<typename Class >
  SEXP Rcpp::internal::make_new_object (Class *ptr)

7.380.1 Macro Definition Documentation

7.380.1.1 #define RcppExport extern "C"

Examples:

ConvolveBenchmarks/convolve3_cpp.cpp, and ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 129 of file RcppCommon.h.

7.381 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

Functions

- attribute_hidden unsigned long Rcpp::internal::enterRNGScope ()
- attribute_hidden unsigned long Rcpp::internal::exitRNGScope ()
- 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)

7.381.1 Macro Definition Documentation

7.381.1.1 #define COMPILING_RCPP

Definition at line 23 of file api.cpp.

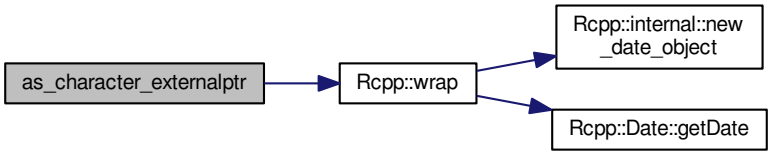
7.381.2 Function Documentation

7.381.2.1 SEXP as_character_externalptr (SEXP *xp*)

Definition at line 150 of file api.cpp.

References Rcpp::wrap().

Here is the call graph for this function:



7.381.2.2 std::string demangle (const std::string & *name*) [inline]

Definition at line 121 of file api.cpp.

Referenced by exception_to_r_condition(), exception_to_try_error(), Rcpp::get_return_type_dispatch(), and register↩ Functions().

7.381.2.3 SEXP rcpp_can_use_cxx0x ( )

Definition at line 237 of file api.cpp.

7.381.2.4 SEXP rcpp_can_use_cxx11 ( )

Definition at line 247 of file api.cpp.

7.381.2.5 SEXP rcpp_capabilities ( )

Definition at line 157 of file api.cpp.

7.381.2.6 const char* short_file_name (const char * *file*)

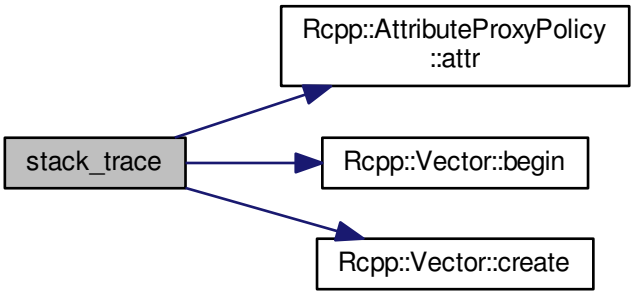
Definition at line 140 of file api.cpp.

7.381.2.7 SEXP stack_trace (const char * *file*, int *line*)

Definition at line 257 of file api.cpp.

References Rcpp::_, Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::create().

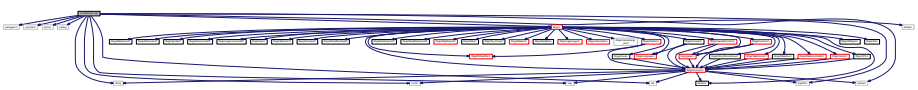
Here is the call graph for this function:



7.382 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::RExportsGenerator
- 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)
- void Rcpp::attributes::showWarning (const std::string &msg)
- bool Rcpp::attributes::isRoxygenCpp (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)
- void Rcpp::attributes::generateCpp (std::ostream &ostr, const SourceFileAttributes &attributes, bool includePrototype, 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 sPlatform)
- RcppExport SEXP compileAttributes (SEXP sPackageDir, SEXP sPackageName, SEXP sDepends, SEXP sCppFiles, SEXP sCppFileBasenames, SEXP sIncludes, SEXP sVerbose, SEXP sPlatform)

Variables

- const char *const Rcpp::attributes::kWhitespaceChars = " \f\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::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"

7.382.1 Macro Definition Documentation

7.382.1.1 #define COMPILING_RCPP

Definition at line 22 of file attributes.cpp.

7.382.1.2 #define RCPP_NO_SUGAR

Definition at line 38 of file attributes.cpp.

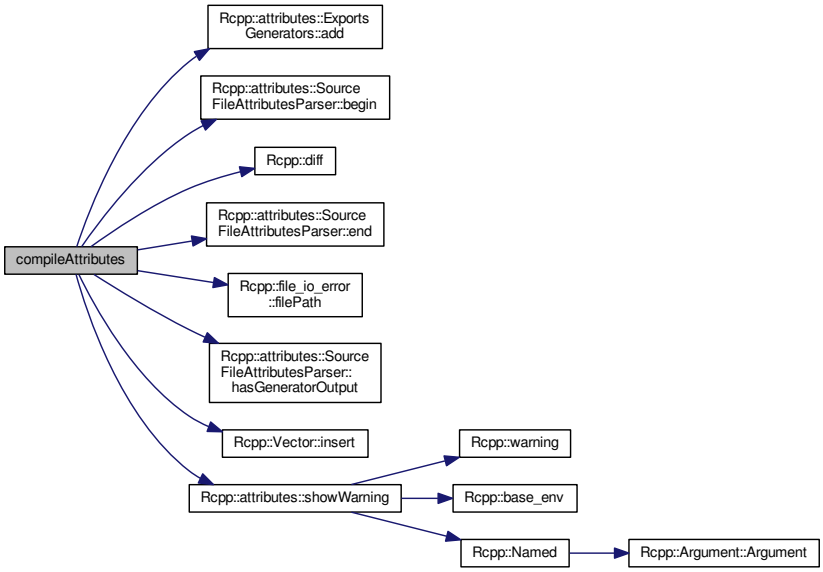
7.382.2 Function Documentation

7.382.2.1 RcppExport SEXP compileAttributes (SEXP *sPackageDir*, SEXP *sPackageName*, SEXP *sDepends*, SEXP *sCppFiles*, SEXP *sCppFileBasenames*, SEXP *sIncludes*, SEXP *sVerbose*, SEXP *sPlatform*)

Definition at line 3292 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerators::add(), Rcpp::attributes::SourceFileAttributesParser::begin(), BEGIN_RCPP, Rcpp::diff(), Rcpp::attributes::SourceFileAttributesParser::end(), END_RCPP, Rcpp::file_io_error::filePath(), Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput(), Rcpp::Vector< RTYPE, StoragePolicy >::insert(), Rcpp::attributes::kDependsAttribute, Rcpp::Rcout, and Rcpp::attributes::showWarning().

Here is the call graph for this function:



7.382.2.2 RcppExport SEXP sourceCppContext (SEXP *sFile,* SEXP *sCode,* SEXP *sRebuild,* SEXP *sCacheDir,* SEXP *sPlatform*)

Definition at line 3228 of file attributes.cpp.

References Rcpp::_, BEGIN_RCPP, Rcpp::Vector< RTYPE, StoragePolicy >::create(), and END_RCPP.

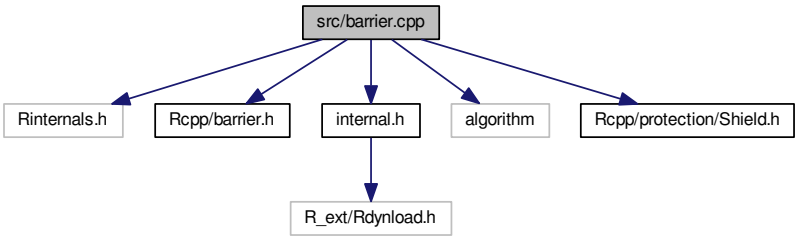
Here is the call graph for this function:

7.383 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 Rcpp::Rcpp_eval (SEXP expr, SEXP env)
- SEXP get_string_elt (SEXP x, int i)
- const char * char_get_string_elt (SEXP x, int i)
- void set_string_elt (SEXP x, int i, SEXP value)
- void char_set_string_elt (SEXP x, int i, const char *value)
- SEXP * get_string_ptr (SEXP x)
- SEXP get_vector_elt (SEXP x, int i)
- void set_vector_elt (SEXP x, int i, SEXP value)
- SEXP * get_vector_ptr (SEXP x)
- void * dataptr (SEXP x)
- const char * char_nocheck (SEXP x)
- 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

7.383.1 Macro Definition Documentation

7.383.1.1 #define COMPILING_RCPP

Definition at line 22 of file barrier.cpp.

7.383.1.2 #define RCPP_CACHE_SIZE 5

Definition at line 87 of file barrier.cpp.

Referenced by init_Rcpp_cache().

7.383.1.3 #define RCPP_HASH_CACHE_INDEX 4

Definition at line 86 of file barrier.cpp.

Referenced by get_cache(), and init_Rcpp_cache().

7.383.1.4 #define RCPP_HASH_CACHE_INITIAL_SIZE 1024

Definition at line 90 of file barrier.cpp.

Referenced by init_Rcpp_cache().

7.383.1.5 #define USE_RINTERNALS

Definition at line 24 of file barrier.cpp.

7.383.2 Function Documentation

7.383.2.1 const char* char_get_string_elt (SEXP *x*, int *i*) [inline]

Definition at line 39 of file barrier.cpp.

Referenced by registerFunctions().

7.383.2.2 const char* char_nocheck (SEXP *x*) [inline]

Definition at line 79 of file barrier.cpp.

Referenced by registerFunctions(), and Rcpp::String::setBuffer().

7.383.2.3 void char_set_string_elt (SEXP *x*, int *i*, const char * *value*) [inline]

Definition at line 49 of file barrier.cpp.

Referenced by registerFunctions().

7.383.2.4 void* dataptr (SEXP *x*) [inline]

Definition at line 74 of file barrier.cpp.

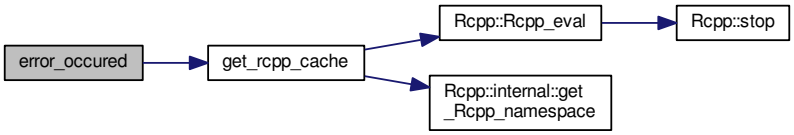
Referenced by registerFunctions().

7.383.2.5 int error_occured ( )

Definition at line 174 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:

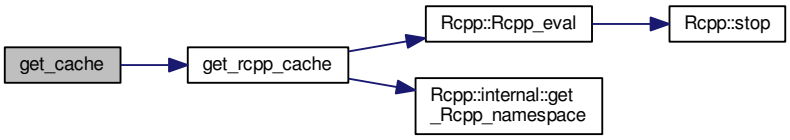


7.383.2.6 int* get_cache (int *m*)

Definition at line 198 of file barrier.cpp.

References get_rcpp_cache(), and RCPP_HASH_CACHE_INDEX.

Here is the call graph for this function:



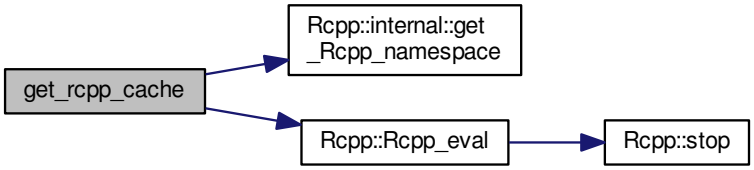
7.383.2.7 SEXP get_rcpp_cache ( )

Definition at line 94 of file barrier.cpp.

References Rcpp::internal::get_Rcpp_namespace(), Rcpp_cache, Rcpp_cache_know, and Rcpp::Rcpp_eval().

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().

Here is the call graph for this function:



7.383.2.8 SEXP get_string_elt (SEXP *x*, int *i*) [inline]

Definition at line 34 of file barrier.cpp.

Referenced by registerFunctions().

7.383.2.9 SEXP* get_string_ptr (SEXP *x*) [inline]

Definition at line 54 of file barrier.cpp.

Referenced by Rcpp::sugar::get_const_begin(), and registerFunctions().

7.383.2.10 SEXP get_vector_elt (SEXP *x*, int *i*) [inline]

Definition at line 59 of file barrier.cpp.

Referenced by registerFunctions().

7.383.2.11 SEXP* get_vector_ptr (SEXP *x*) [inline]

Definition at line 69 of file barrier.cpp.

Referenced by registerFunctions().

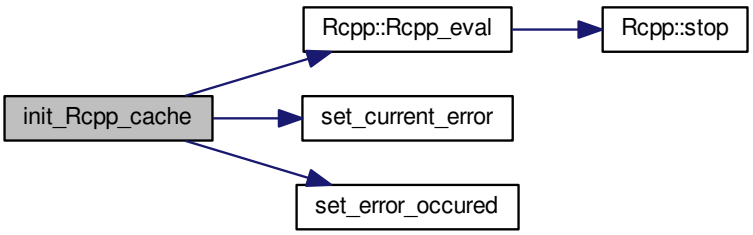
7.383.2.12 SEXP init_Rcpp_cache ( )

Definition at line 138 of file barrier.cpp.

References RCPP_CACHE_SIZE, Rcpp::Rcpp_eval(), RCPP_HASH_CACHE_INDEX, RCPP_HASH_CACHE_INIT↩IAL_SIZE, set_current_error(), and set_error_occured().

Referenced by R_init_Rcpp().

Here is the call graph for this function:

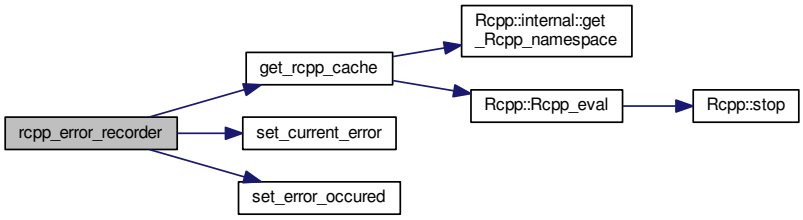


7.383.2.13 SEXP rcpp_error_recorder (SEXP *e*)

Definition at line 180 of file barrier.cpp.

References get_rcpp_cache(), set_current_error(), and set_error_occured().

Here is the call graph for this function:

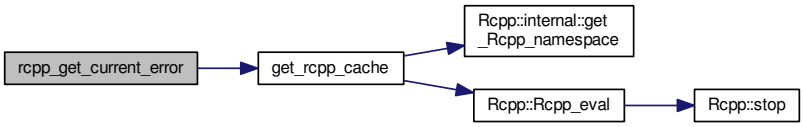


7.383.2.14 SEXP rcpp_get_current_error ( )

Definition at line 193 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:

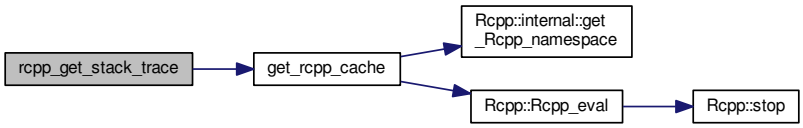


7.383.2.15 SEXP rcpp_get_stack_trace ( )

Definition at line 118 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:

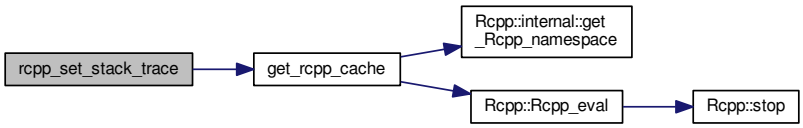


7.383.2.16 SEXP rcpp_set_stack_trace (SEXP *e*)

Definition at line 123 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:

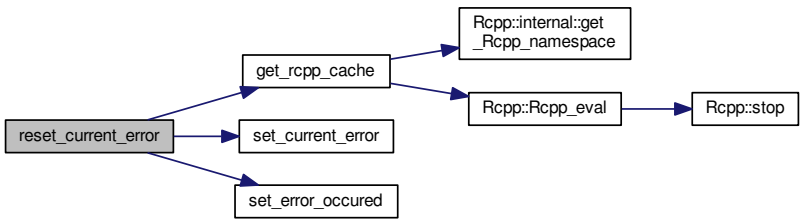


7.383.2.17 SEXP reset_current_error ( )

Definition at line 158 of file barrier.cpp.

References get_rcpp_cache(), set_current_error(), and set_error_occured().

Here is the call graph for this function:



7.383.2.18 SEXP set_current_error (SEXP *cache,* SEXP *e*)

Definition at line 133 of file barrier.cpp.

Referenced by init_Rcpp_cache(), rcpp_error_recorder(), and reset_current_error().

7.383.2.19 SEXP set_error_occured (SEXP *cache,* SEXP *e*)

Definition at line 128 of file barrier.cpp.

Referenced by init_Rcpp_cache(), rcpp_error_recorder(), and reset_current_error().

7.383.2.20 void set_string_elt (SEXP *x,* int *i,* SEXP *value*) [inline]

Definition at line 44 of file barrier.cpp.

Referenced by registerFunctions().

7.383.2.21 void set_vector_elt (SEXP *x,* int *i,* SEXP *value*) [inline]

Definition at line 64 of file barrier.cpp.

Referenced by registerFunctions().

7.383.3 Variable Documentation

7.383.3.1 SEXP Rcpp_cache = R_NilValue [static]

Definition at line 84 of file barrier.cpp.

Referenced by get_rcpp_cache().

7.383.3.2 bool Rcpp_cache_know = false [static]

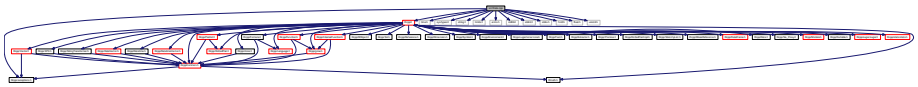
Definition at line 83 of file barrier.cpp.

Referenced by get_rcpp_cache().

7.384 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::ttinfo
- 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 GRANDPARENTED "Local time zone must be set--see zic manual page"
- #define YEARSPERREPEAT 400 /* years before a Gregorian repeat */
- #define AVGSECSPERYEAR 31556952L

- #define SECSPERREPEAT ((int_fast64_t) YEARSPERREPEAT * (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 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 50 /* 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 ((long) 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 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, long ∗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 time_t Rcpp::transtime (const time_t janfirst, const int year, const struct rule ∗const rulep, const long offset)
- static struct tm ∗ Rcpp::timesub (const time_t ∗const timep, const long offset, const struct state ∗const sp, struct tm ∗const tmp)
- static int Rcpp::leaps_thru_end_of (const int y)
- static int Rcpp::increment_overflow (int ∗number, int delta)
- static long Rcpp::detzcode (const char ∗const codep)
- static time_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, long ∗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 long offset, struct tm ∗const tmp)
- attribute_hidden struct tm ∗ Rcpp::gmtime_ (const time_t ∗const x)

Variables

- 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

7.384.1 Macro Definition Documentation

7.384.1.1 #define _NO_OLDNAMES /∗ avoid tznames ∗/

Definition at line 89 of file Date.cpp.

7.384.1.2 #define AVGSECSPERYEAR 31556952L

Definition at line 113 of file Date.cpp.

7.384.1.3 #define BIGGEST(*a*, *b*) (((a) > (b)) ? (a) : (b))

Definition at line 312 of file Date.cpp.

7.384.1.4 #define COMPILING_RCPP

Definition at line 29 of file Date.cpp.

7.384.1.5 #define DAY_OF_YEAR 1 /* n - day of year */

Definition at line 358 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

7.384.1.6 #define days_in_year(*year*) (isleap(year) ? 366 : 365)

7.384.1.7 #define DAYSPERLYEAR 366

Definition at line 243 of file Date.cpp.

Referenced by Rcpp::timesub().

7.384.1.8 #define DAYSPERNYEAR 365

Definition at line 242 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::timesub().

7.384.1.9 #define DAYSPERWEEK 7

Definition at line 241 of file Date.cpp.

Referenced by Rcpp::getrule(), Rcpp::getsecs(), Rcpp::timesub(), and Rcpp::transtime().

7.384.1.10 #define EOVERFLOW 79

Definition at line 95 of file Date.cpp.

7.384.1.11 #define EPOCH_WDAY TM_THURSDAY

Definition at line 270 of file Date.cpp.

Referenced by Rcpp::timesub().

7.384.1.12 #define EPOCH_YEAR 1970

Definition at line 269 of file Date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzparse().

7.384.1.13 #define gmtptr (&gmtmem)

Definition at line 375 of file Date.cpp.

Referenced by Rcpp::gmtsub().

7.384.1.14 #define GRANDPARENTED "Local time zone must be set--see zic manual page"

Definition at line 111 of file Date.cpp.

7.384.1.15 #define HOURSPERDAY 24

Definition at line 240 of file Date.cpp.

Referenced by Rcpp::getsecs().

7.384.1.16 #define INITIALIZE(*x*) (x = 0)

Definition at line 117 of file Date.cpp.

Referenced by Rcpp::transtime(), and Rcpp::tzparse().

7.384.1.17 #define is_digit(*c*) ((unsigned)(c) - '0' <= 9)

Definition at line 116 of file Date.cpp.

Referenced by Rcpp::getnum(), Rcpp::getrule(), and Rcpp::getzname().

7.384.1.18 #define isleap(*y*) ((((y) % 4) == 0 && ((y) % 100) != 0) || ((y) % 400) == 0)

Definition at line 272 of file Date.cpp.

Referenced by Rcpp::timesub(), Rcpp::transtime(), and Rcpp::tzparse().

7.384.1.19 #define isleap(y) (((y) % 4) == 0 && (((y) % 100) != 0 || ((y) % 400) == 0))

Definition at line 272 of file Date.cpp.

7.384.1.20 #define isleap_sum(a, b) isleap((a) % 400 + (b) % 400)

Definition at line 286 of file Date.cpp.

7.384.1.21 #define JULIAN_DAY 0 /* Jn - Julian day */

Definition at line 357 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

7.384.1.22 #define MINSPERHOUR 60

Definition at line 239 of file Date.cpp.

Referenced by Rcpp::getsecs().

7.384.1.23 #define MONSPERYEAR 12

Definition at line 246 of file Date.cpp.

Referenced by Rcpp::getrule().

7.384.1.24 #define MONTH_NTH_DAY_OF_WEEK 2 /* Mm.n.d - month, week, day of week */

Definition at line 359 of file Date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

7.384.1.25 #define MY_TZNAME_MAX 255

Definition at line 318 of file Date.cpp.

7.384.1.26 #define OPEN_MODE O_RDONLY

Definition at line 296 of file Date.cpp.

Referenced by Rcpp::tzload().

7.384.1.27 #define SECSPERDAY ((long) SECSPERHOUR * HOURSPERDAY)

Definition at line 245 of file Date.cpp.

Referenced by Rcpp::timesub(), Rcpp::transtime(), and Rcpp::tzparse().

7.384.1.28 #define SECSPERHOUR (SECSPERMIN * MINSPERHOUR)

Definition at line 244 of file Date.cpp.

Referenced by Rcpp::getrule(), Rcpp::getsecs(), Rcpp::timesub(), and Rcpp::tzparse().

7.384.1.29 #define SECSPERMIN 60

Definition at line 238 of file Date.cpp.

Referenced by Rcpp::getsecs(), and Rcpp::timesub().

7.384.1.30 #define SECSPERREPEAT ((int_fast64_t) YEARSPERREPEAT * (int_fast64_t) AVGSECSPERYEAR)

Definition at line 114 of file Date.cpp.

Referenced by Rcpp::differ_by_repeat().

7.384.1.31 #define SECSPERREPEAT_BITS 34 /* ceil(log2(SECSPERREPEAT)) */

Definition at line 115 of file Date.cpp.

Referenced by Rcpp::differ_by_repeat().

7.384.1.32 #define TM_APRIL 3

Definition at line 259 of file Date.cpp.

7.384.1.33 #define TM_AUGUST 7

Definition at line 263 of file Date.cpp.

7.384.1.34 #define TM_DECEMBER 11

Definition at line 267 of file Date.cpp.

7.384.1.35 #define TM_FEBRUARY 1

Definition at line 257 of file Date.cpp.

7.384.1.36 #define TM_FRIDAY 5

Definition at line 253 of file Date.cpp.

7.384.1.37 #define TM_JANUARY 0

Definition at line 256 of file Date.cpp.

7.384.1.38 #define TM_JULY 6

Definition at line 262 of file Date.cpp.

7.384.1.39 #define TM_JUNE 5

Definition at line 261 of file Date.cpp.

7.384.1.40 #define TM_MARCH 2

Definition at line 258 of file Date.cpp.

7.384.1.41 #define TM_MAY 4

Definition at line 260 of file Date.cpp.

7.384.1.42 #define TM_MONDAY 1

Definition at line 249 of file Date.cpp.

7.384.1.43 #define TM_NOVEMBER 10

Definition at line 266 of file Date.cpp.

7.384.1.44 #define TM_OCTOBER 9

Definition at line 265 of file Date.cpp.

7.384.1.45 #define TM_SATURDAY 6

Definition at line 254 of file Date.cpp.

7.384.1.46 #define TM_SEPTEMBER 8

Definition at line 264 of file Date.cpp.

7.384.1.47 #define TM_SUNDAY 0

Definition at line 248 of file Date.cpp.

7.384.1.48 #define TM_THURSDAY 4

Definition at line 252 of file Date.cpp.

7.384.1.49 #define TM_TUESDAY 2

Definition at line 250 of file Date.cpp.

7.384.1.50 #define TM_WEDNESDAY 3

Definition at line 251 of file Date.cpp.

7.384.1.51 #define TYPE_BIT(*type*) (sizeof (type) * CHAR_BIT)

Definition at line 108 of file Date.cpp.

Referenced by Rcpp::differ_by_repeat().

7.384.1.52 #define TYPE_INTEGRAL(*type*) (((type) 0.5) != 0.5)

Definition at line 110 of file Date.cpp.

Referenced by Rcpp::differ_by_repeat(), and Rcpp::tzload().

7.384.1.53 #define TYPE_SIGNED(*type*) (((type) -1) < 0)

Definition at line 109 of file Date.cpp.

Referenced by Rcpp::differ_by_repeat(), and Rcpp::tzload().

7.384.1.54 #define TZ_MAGIC "TZif"

Definition at line 158 of file Date.cpp.

7.384.1.55 #define TZ_MAX_CHARS 50 /∗ Maximum number of abbreviation characters ∗/

Definition at line 230 of file Date.cpp.

Referenced by Rcpp::tzload().

7.384.1.56 #define TZ_MAX_LEAPS 50 /∗ Maximum number of leap second corrections ∗/

Definition at line 235 of file Date.cpp.

Referenced by Rcpp::tzload().

7.384.1.57 #define TZ_MAX_TIMES 1200

Definition at line 213 of file Date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

7.384.1.58 #define TZ_MAX_TYPES 256 /∗ Limited by what (unsigned char)'s can hold ∗/

Definition at line 218 of file Date.cpp.

Referenced by Rcpp::tzload().

7.384.1.59 #define TZDEFAULT "localtime"

Definition at line 147 of file Date.cpp.

Referenced by Rcpp::tzload().

7.384.1.60 #define TZDEFRULES "America/New_York"

Definition at line 151 of file Date.cpp.

Referenced by Rcpp::tzparse().

7.384.1.61 #define TZDEFRULESTRING ",M4.1.0,M10.5.0"

Definition at line 309 of file Date.cpp.

Referenced by Rcpp::tzparse().

7.384.1.62 #define TZDIR "/usr/local/etc/zoneinfo" /* Time zone object file directory */

Definition at line 143 of file Date.cpp.

7.384.1.63 #define TZFILE_H

Definition at line 123 of file Date.cpp.

7.384.1.64 #define YEARSPERREPEAT 400 /* years before a Gregorian repeat */

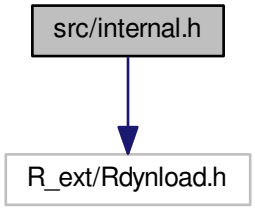
Definition at line 112 of file Date.cpp.

Referenced by Rcpp::tzload().

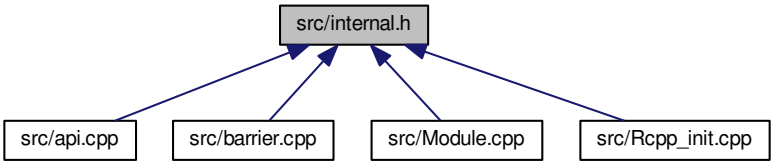
7.385 src/internal.h File Reference

```
#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 (CppObject__finalize)
- CALLFUN_0 (get_rcpp_cache)

- CALLFUN_0 (init_Rcpp_cache)
- CALLFUN_1 (rcpp_error_recorder)
- CALLFUN_3 (CppField__get)
- CALLFUN_4 (CppField__set)
- CALLFUN_0 (rcpp_capabilities)
- CALLFUN_0 (rcpp_can_use_cxx0x)
- CALLFUN_0 (rcpp_can_use_cxx11)
- EXTFUN (CppMethod__invoke)
- EXTFUN (CppMethod__invoke_void)
- EXTFUN (CppMethod__invoke_notvoid)
- EXTFUN (InternalFunction_invoke)
- EXTFUN (Module__invoke)
- EXTFUN (class__newInstance)
- EXTFUN (class__dummyInstance)
- void init_Rcpp_routines (DllInfo *)

7.385.1 Macro Definition Documentation

7.385.1.1 #define CALLFUN_0(*name*) SEXP name()

Definition at line 27 of file internal.h.

7.385.1.2 #define CALLFUN_1(*name*) SEXP name(SEXP)

Definition at line 28 of file internal.h.

7.385.1.3 #define CALLFUN_2(*name*) SEXP name(SEXP,SEXP)

Definition at line 29 of file internal.h.

7.385.1.4 #define CALLFUN_3(*name*) SEXP name(SEXP,SEXP,SEXP)

Definition at line 30 of file internal.h.

7.385.1.5 #define CALLFUN_4(*name*) SEXP name(SEXP,SEXP,SEXP,SEXP)

Definition at line 31 of file internal.h.

7.385.1.6 #define CALLFUN_5(*name*) SEXP name(SEXP,SEXP,SEXP,SEXP,SEXP)

Definition at line 32 of file internal.h.

7.385.1.7 #define EXTFUN(*name*) SEXP name(SEXP)

Definition at line 33 of file internal.h.

7.385.1.8 #define MAX_ARGS 65

Definition at line 38 of file internal.h.

7.385.1.9 #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.

7.385.1.10 #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.

7.385.1.11 #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.

7.385.1.12 #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.

7.385.1.13 #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.

Referenced by class__newInstance(), CppMethod__invoke(), CppMethod__invoke_notvoid(), CppMethod__invoke_↩ void(), InternalFunction_invoke(), and Module__invoke().

7.385.2 Function Documentation

7.385.2.1 CALLFUN_0 (get_rcpp_cache)

7.385.2.2 CALLFUN_0 (init_Rcpp_cache)

7.385.2.3 CALLFUN_0 (rcpp_capabilities)

7.385.2.4 CALLFUN_0 (rcpp_can_use_cxx0x)

7.385.2.5 CALLFUN_0 (rcpp_can_use_cxx11)

7.385.2.6 CALLFUN_1 (as_character_externalptr)

7.385.2.7 CALLFUN_1 (Class__name)

7.385.2.8 CALLFUN_1 (Class__has_default_constructor)

7.385.2.9 CALLFUN_1 (CppClass__complete)

7.385.2.10 CALLFUN_1 (CppClass__methods)

7.385.2.11 CALLFUN_1 (Module__classes_info)

7.385.2.12 CALLFUN_1 (Module__complete)

7.385.2.13 CALLFUN_1 (Module__functions_arity)

7.385.2.14 CALLFUN_1 (Module__functions_names)

7.385.2.15 CALLFUN_1 (Module__name)

7.385.2.16 CALLFUN_1 (rcpp_error_recorder)

7.385.2.17 CALLFUN_2 (Module__get_class)

7.385.2.18 CALLFUN_2 (Module__has_class)

7.385.2.19 CALLFUN_2 (Module__has_function)

7.385.2.20 CALLFUN_2 (Module__get_function)

7.385.2.21 CALLFUN_2 (CppObject__finalize)

7.385.2.22 CALLFUN_3 (CppField__get)

7.385.2.23 CALLFUN_4 (CppField__set)

7.385.2.24 EXTFUN (CppMethod__invoke)

7.385.2.25 EXTFUN (CppMethod__invoke_void)

7.385.2.26 EXTFUN (CppMethod__invoke_notvoid)

7.385.2.27 EXTFUN (InternalFunction_invoke)

7.385.2.28 EXTFUN (Module__invoke)

7.385.2.29 EXTFUN (class__newInstance)

7.385.2.30 EXTFUN (class__dummyInstance)

7.385.2.31 SEXP get_Rcpp_protection_stack ( )

7.385.2.32 void init_Rcpp_routines (DllInfo *)

Definition at line 76 of file Rcpp_init.cpp.

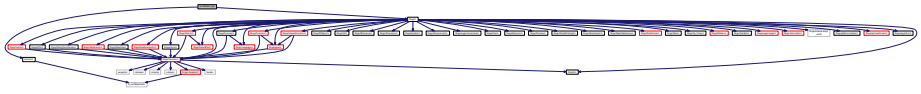
References callEntries, and extEntries.

Referenced by R_init_Rcpp().

7.386 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::CppFunctionBase > 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, CppField__get, XP_Class cl, SEXP field_xp, SEXP obj)
- RCPP_FUN_4 (SEXP, CppField__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

7.386.1 Macro Definition Documentation

7.386.1.1 #define CHECK_DUMMY_OBJ(*p*) if (p == rcpp_dummy_pointer) throw Rcpp::not_initialized()

Definition at line 149 of file Module.cpp.

Referenced by CppMethod__invoke(), CppMethod__invoke_notvoid(), and CppMethod__invoke_void().

7.386.1.2 #define COMPILING_RCPP

Definition at line 22 of file Module.cpp.

7.386.2 Typedef Documentation

7.386.2.1 typedef Rcpp::XPtr<Rcpp::class_Base> XP_Class

Definition at line 28 of file Module.cpp.

7.386.2.2 typedef Rcpp::XPtr<Rcpp::CppFunctionBase> XP_Function

Definition at line 29 of file Module.cpp.

7.386.2.3 typedef Rcpp::XPtr<Rcpp::Module> XP_Module

Definition at line 27 of file Module.cpp.

7.386.3 Function Documentation

7.386.3.1 SEXP class__dummyInstance (SEXP *args*)

Definition at line 152 of file Module.cpp.

References rcpp_dummy_pointer.

7.386.3.2 SEXP class__newInstance (SEXP *args*)

Definition at line 137 of file Module.cpp.

References UNPACK_EXTERNAL_ARGS.

7.386.3.3 SEXP CppMethod__invoke (SEXP *args*)

Definition at line 166 of file Module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.386.3.4 SEXP CppMethod__invoke_notvoid (SEXP *args*)

Definition at line 204 of file Module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.386.3.5 SEXP CppMethod__invoke_void (SEXP *args*)

Definition at line 185 of file Module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.386.3.6 Rcpp::Module* getCurrentScope ( )

Definition at line 227 of file Module.cpp.

References Rcpp::current_scope.

7.386.3.7 SEXP InternalFunction_invoke (SEXP *args*)

Definition at line 117 of file Module.cpp.

References BEGIN_RCPP, END_RCPP, and UNPACK_EXTERNAL_ARGS.

7.386.3.8 SEXP Module__invoke (SEXP *args*)

Definition at line 126 of file Module.cpp.

References BEGIN_RCPP, END_RCPP, and UNPACK_EXTERNAL_ARGS.

7.386.3.9 RCPP_FUN_1 (bool , Class__has_default_constructor , XP_Class *cl*)

Definition at line 31 of file Module.cpp.

7.386.3.10 RCPP_FUN_1 (std::string , Class__name , XP_Class *cl*)

Definition at line 43 of file Module.cpp.

7.386.3.11 RCPP_FUN_1 (bool , CppObject__needs_init , SEXP *xp*)

Definition at line 55 of file Module.cpp.

7.386.3.12 RCPP_FUN_1 (Rcpp::CharacterVector , CppClass__methods , XP_Class *cl*)

Definition at line 58 of file Module.cpp.

7.386.3.13 RCPP_FUN_1 (Rcpp::CharacterVector , CppClass__properties , XP_Class *cl*)

Definition at line 61 of file Module.cpp.

7.386.3.14 RCPP_FUN_1 (Rcpp::List , CppClass__property_classes , XP_Class *cl*)

Definition at line 64 of file Module.cpp.

7.386.3.15 RCPP_FUN_1 (Rcpp::IntegerVector , CppClass__methods_arity , XP_Class *cl*)

Definition at line 68 of file Module.cpp.

7.386.3.16 RCPP_FUN_1 (Rcpp::LogicalVector , CppClass__methods_voidness , XP_Class *cl*)

Definition at line 71 of file Module.cpp.

7.386.3.17 RCPP_FUN_1 (Rcpp::IntegerVector , Module__functions_arity , XP_Module *module*)

Definition at line 83 of file Module.cpp.

7.386.3.18 RCPP_FUN_1 (Rcpp::CharacterVector , Module__functions_names , XP_Module *module*)

Definition at line 86 of file Module.cpp.

7.386.3.19 RCPP_FUN_1 (std::string , Module__name , XP_Module *module*)

Definition at line 89 of file Module.cpp.

7.386.3.20 RCPP_FUN_1 (Rcpp::List , Module__classes_info , XP_Module *module*)

Definition at line 92 of file Module.cpp.

7.386.3.21 RCPP_FUN_1 (Rcpp::CharacterVector , Module__complete , XP_Module *module*)

Definition at line 95 of file Module.cpp.

7.386.3.22 RCPP_FUN_1 (Rcpp::CharacterVector , CppClass__complete , XP_Class *cl*)

Definition at line 98 of file Module.cpp.

7.386.3.23 RCPP_FUN_2 (SEXP , Module__get_function , XP_Module *module,* std::string *fun*)

Definition at line 34 of file Module.cpp.

7.386.3.24 RCPP_FUN_2 (bool , Class__has_method , XP_Class *cl,* std::string *m*)

Definition at line 37 of file Module.cpp.

7.386.3.25 RCPP_FUN_2 (bool , Class__has_property , XP_Class *cl,* std::string *m*)

Definition at line 40 of file Module.cpp.

7.386.3.26 RCPP_FUN_2 (bool , Module__has_function , XP_Module *module,* std::string *met*)

Definition at line 46 of file Module.cpp.

7.386.3.27 RCPP_FUN_2 (bool , Module__has_class , XP_Module *module,* std::string *cl*)

Definition at line 49 of file Module.cpp.

7.386.3.28 RCPP_FUN_2 (Rcpp::CppClass , Module__get_class , XP_Module *module,* std::string *cl*)

Definition at line 52 of file Module.cpp.

7.386.3.29 RCPP_FUN_2 (bool , CppClass__property_is_readonly , XP_Class *cl*, std::string *p*)

Definition at line 76 of file Module.cpp.

7.386.3.30 RCPP_FUN_2 (std::string , CppClass__property_class , XP_Class *cl*, std::string *p*)

Definition at line 79 of file Module.cpp.

7.386.3.31 RCPP_FUN_2 (SEXP , CppObject__finalize , XP_Class *cl*, SEXP *obj*)

Definition at line 111 of file Module.cpp.

7.386.3.32 RCPP_FUN_3 (SEXP , CppField__get , XP_Class *cl*, SEXP *field_xp*, SEXP *obj*)

Definition at line 104 of file Module.cpp.

7.386.3.33 RCPP_FUN_4 (SEXP , CppField__set , XP_Class *cl*, SEXP *field_xp*, SEXP *obj*, SEXP *value*)

Definition at line 107 of file Module.cpp.

7.386.3.34 void setCurrentScope (Rcpp::Module * *scope*)

Definition at line 230 of file Module.cpp.

References Rcpp::current_scope.

7.386.4 Variable Documentation

7.386.4.1 SEXP rcpp_dummy_pointer = R_NilValue

Definition at line 147 of file Module.cpp.

Referenced by class__dummyInstance().

7.387 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_RegisterCCallable("Rcpp", #__FUN__ , (DL_FUNC)__FUN__);

Functions

- void init_Rcpp_routines (DllInfo *info)
- void registerFunctions ()
- void R_unload_Rcpp (DllInfo *info)
- void R_init_Rcpp (DllInfo *info)

Variables

- static R_CallMethodDef callEntries [ ]
- static R_ExternalMethodDef extEntries [ ]

7.387.1 Macro Definition Documentation

7.387.1.1 #define CALLDEF(*name, n*) {#name, (DL_FUNC) &name, n}

Definition at line 28 of file Rcpp_init.cpp.

7.387.1.2 #define COMPILING_RCPP

Definition at line 22 of file Rcpp_init.cpp.

7.387.1.3 #define EXTDEF(*name*) {#name, (DL_FUNC) &name, -1}

Definition at line 29 of file Rcpp_init.cpp.

7.387.1.4 #define RCPP_REGISTER(*__FUN__*) R_RegisterCCallable("Rcpp", #__FUN__ , (DL_FUNC)__FUN__);

Referenced by registerFunctions().

7.387.2 Function Documentation

7.387.2.1 void init_Rcpp_routines (DllInfo * *info*)

Definition at line 76 of file Rcpp_init.cpp.

References callEntries, and extEntries.

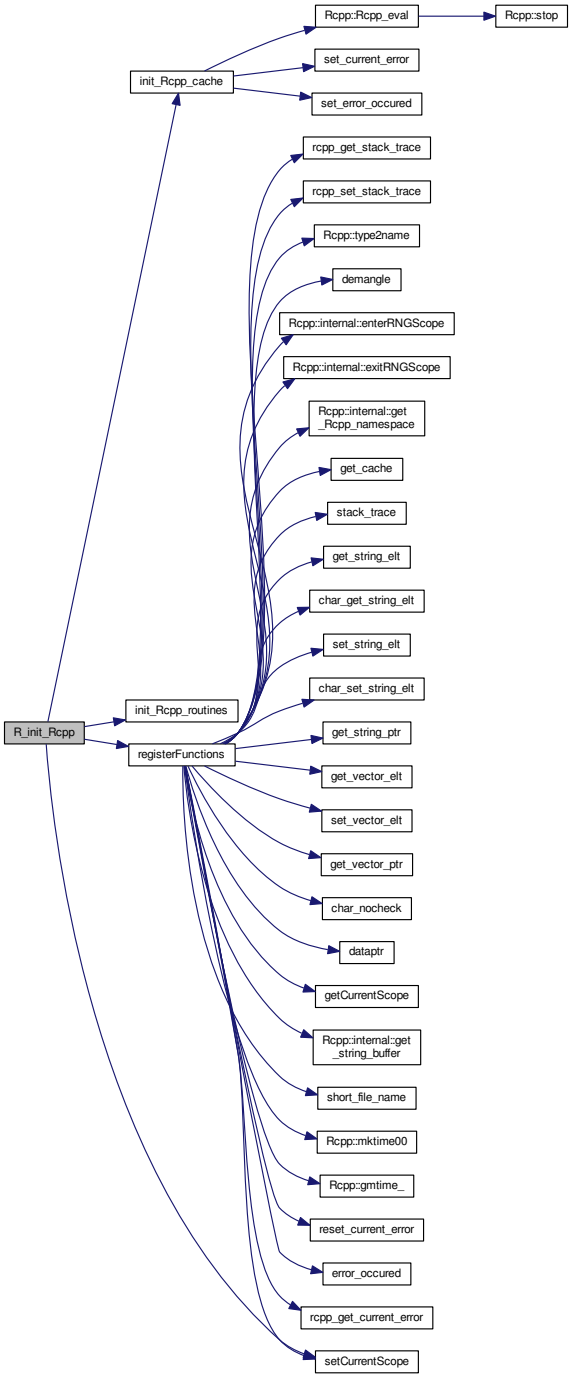
Referenced by R_init_Rcpp().

7.387.2.2 void R_init_Rcpp (DllInfo * *info*)

Definition at line 128 of file Rcpp_init.cpp.

References init_Rcpp_cache(), init_Rcpp_routines(), registerFunctions(), and setCurrentScope().

Here is the call graph for this function:



7.387.2.3 void R_unload_Rcpp (DllInfo * *info*)

Definition at line 124 of file Rcpp_init.cpp.

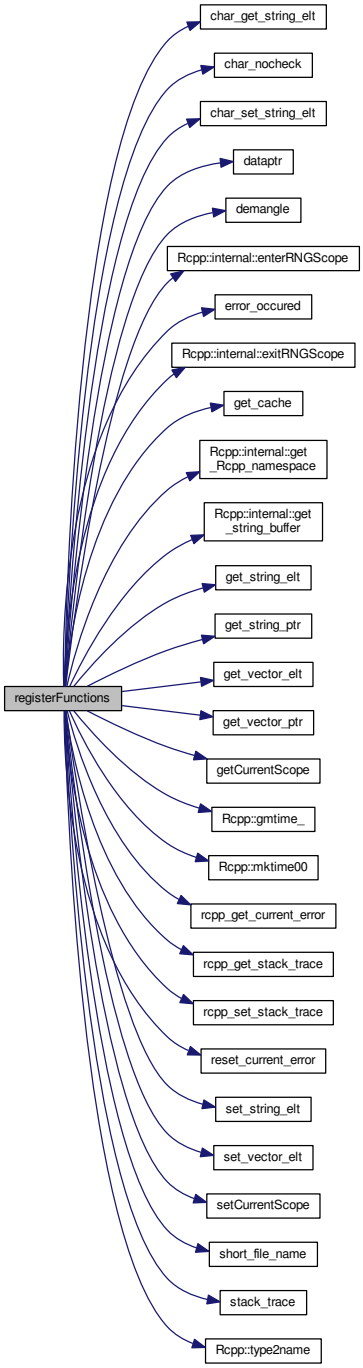
7.387.2.4 void registerFunctions ( )

Definition at line 86 of file Rcpp_init.cpp.

References char_get_string_elt(), char_nocheck(), char_set_string_elt(), dataptr(), demangle(), 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_get_current_error(), rcpp_get_stack_trace(), RCPP_REGISTER, rcpp_set_stack_trace(), reset_current_error(), set_string_elt(), set_vector_elt(), setCurrentScope(), short_file_name(), stack_trace(), and Rcpp::type2name().

Referenced by R_init_Rcpp().

Here is the call graph for this function:



7.387.3 Variable Documentation

7.387.3.1 R_CallMethodDef callEntries[ ] [static]

Definition at line 31 of file Rcpp_init.cpp.

Referenced by init_Rcpp_routines().

7.387.3.2 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 63 of file Rcpp_init.cpp.

Referenced by init_Rcpp_routines().

Chapter 8

Example Documentation

8.1 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)
```

8.2 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>.

8.3 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)
```

8.4 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)
```

8.5 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)
```

8.6 ConvolveBenchmarks/exampleRCode.r

R file / littler script to run and time the various implementations.

```
1 #!/usr/bin/env r
2
3 suppressMessages(require(Rcpp))
4 set.seed(42)
5 n <- 200
6 a <- rnorm(n)
7 b <- rnorm(n)
8
9 ## load shared libraries with wrapper code
10 dyn.load("convolve2_c.so")
11 dyn.load("convolve3_cpp.so")
12 dyn.load("convolve4_cpp.so")
13 dyn.load("convolve5_cpp.so")
14 dyn.load("convolve7_c.so")
15
16 dyn.load("convolve8_cpp.so")
17 dyn.load("convolve9_cpp.so")
18 dyn.load("convolve10_cpp.so")
19 dyn.load("convolve11_cpp.so")
20 dyn.load("convolve12_cpp.so" )
21 dyn.load("convolve14_cpp.so" )
22
23 ## now run each one once for comparison of results,
24 ## and define test functions
25
26 R_API_optimised <- function(n,a,b) .Call("convolve2__loop", n, a, b)
27 Rcpp_New_std <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b)
28 #Rcpp_New_std_inside <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b, PACKAGE = "Rcpp" )
29 Rcpp_New_ptr <- function(n,a,b) .Call("convolve4cpp__loop", n, a, b)
30 Rcpp_New_sugar <- function(n,a,b) .Call("convolve5cpp__loop", n, a, b)
31 Rcpp_New_sugar_noNA <- function(n,a,b) .Call("convolve11cpp__loop", n, a, b)
```

```
32 R_API_naive <- function(n,a,b) .Call("convolve7__loop", n, a, b)
33 Rcpp_New_std_2 <- function(n,a,b) .Call("convolve8cpp__loop", n, a, b)
34 #Rcpp_New_std_3 <- function(n,a,b) .Call("convolve9cpp__loop", n, a, b)
35 #Rcpp_New_std_4 <- function(n,a,b) .Call("convolve10cpp__loop", n, a, b)
36 Rcpp_New_std_it <- function(n,a,b) .Call("convolve12cpp__loop", n, a, b )
37 Rcpp_New_std_Fast <- function(n,a,b) .Call("convolve14cpp__loop", n, a, b )
38
39
40 v1 <- R_API_optimised(1L, a, b )
41 v3 <- Rcpp_New_std(1L, a, b)
42 v4 <- Rcpp_New_ptr(1L, a, b)
43 v5 <- Rcpp_New_sugar(1L, a, b )
44 v7 <- R_API_naive(1L, a, b)
45 v11 <- Rcpp_New_sugar_noNA(1L, a, b)
46
47 stopifnot(all.equal(v1, v3))
48 stopifnot(all.equal(v1, v4))
49 stopifnot(all.equal(v1, v5))
50 stopifnot(all.equal(v1, v7))
51 stopifnot(all.equal(v1, v11))
52
53 ## load benchmarkin helper function
54 suppressMessages(library(rbenchmark))
55 REPS <- 5000L
56 bm <- benchmark(R_API_optimised(REPS,a,b),
57                 R_API_naive(REPS,a,b),
58                 Rcpp_New_std(REPS,a,b),
59 #                 Rcpp_New_std_inside(REPS,a,b),
60                 Rcpp_New_ptr(REPS,a,b),
61                 Rcpp_New_sugar(REPS,a,b),
62                 Rcpp_New_sugar_noNA(REPS,a,b),
63                 Rcpp_New_std_2(REPS,a,b),
64 #                 Rcpp_New_std_3(REPS,a,b),
65 #                 Rcpp_New_std_4(REPS,a,b),
66                 Rcpp_New_std_it(REPS,a,b),
67                 Rcpp_New_std_Fast(REPS,a,b),
68                 columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
69                 order="relative",
70                 replications=1)
71 print(bm)
72
73 cat("All results are equal\n") # as we didn't get stopped
74 q("no")
75
76
77 sizes <- 1:10*100
78 REPS <- 5000L
79 timings <- lapply( sizes, function(size){
80     cat( "size = ", size, "..." )
81     a <- rnorm(size); b <- rnorm(size)
82     bm <- benchmark(R_API_optimised(REPS,a,b),
83                 R_API_naive(REPS,a,b),
84                 Rcpp_New_std(REPS,a,b),
85                 Rcpp_New_ptr(REPS,a,b),
86                 Rcpp_New_sugar(REPS,a,b),
87                 Rcpp_New_sugar_noNA(REPS,a,b),
88                 columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
89                 order="relative",
90                 replications=1)
91
92      cat( "  done\n" )
93      bm
94 } )
95 for( i in seq_along(sizes)){
96     timings[[i]]$size <- sizes[i]
97 }
98 timings <- do.call( rbind, timings )
99
100 require( lattice )
101 png( "elapsed.png", width = 800, height = 600 )
102 xyplot( elapsed ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
103 dev.off()
104 png( "relative.png", width = 800, height = 600 )
105 xyplot( relative ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
106 dev.off()
107
```

8.7 FastLM/benchmark.r

Linear model benchmark master file

```
1 #!/usr/bin/env r
2 #
3 # Comparison benchmark
4 #
5 # This shows how Armadillo improves on the previous version using GNU GSL,
6 # and how both are doing better than lm.fit()
7 #
8 # Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
9 #
10 # This file is part of Rcpp.
11 #
12 # Rcpp is free software: you can redistribute it and/or modify it
13 # under the terms of the GNU General Public License as published by
14 # the Free Software Foundation, either version 2 of the License, or
15 # (at your option) any later version.
16 #
17 # Rcpp is distributed in the hope that it will be useful, but
18 # WITHOUT ANY WARRANTY; without even the implied warranty of
19 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
20 # GNU General Public License for more details.
21 #
22 # You should have received a copy of the GNU General Public License
23 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
24
25 suppressMessages(library(RcppGSL))
26 suppressMessages(library(RcppArmadillo))
27
28 source("lmArmadillo.R")
29 source("lmGSL.R")
30
31 set.seed(42)
32 n <- 5000
33 k <- 9
34 X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
35 truecoef <- 1:(k+1)
36 y <- as.numeric(X %*% truecoef + rnorm(n))
37
38 N <- 100
39
40 lmgsl <- lmGSL()
41 lmarma <- lmArmadillo()
42
43 tlm <- mean(replicate(N, system.time( lmfit <- lm(y ~ X - 1) )["elapsed"]), trim=0.05)
44 tlmfit <- mean(replicate(N, system.time(lmfitfit <- lm.fit(X, y))["elapsed"]), trim=0.05)
45 tlmgsl <- mean(replicate(N, system.time(lmgsl(y, X))["elapsed"]), trim=0.05)
46 tlmarma <- mean(replicate(N, system.time(lmarma(y, X))["elapsed"]), trim=0.05)
47
48 res <- c(tlm, tlmfit, tlmgsl, tlmarma)
49 data <- data.frame(results=res, ratios=tlm/res)
50 rownames(data) <- c("lm", "lm.fit", "lmGSL", "lmArma")
51 cat("For n=", n, " and k=", k, "\n", sep="")
52 print(t(data))
53 print(t(1/data[,1,drop=FALSE])) # regressions per second
54
```

8.8 FastLM/fastLMviaArmadillo.r

Runs lm via Armadillo and times the run

```
1 #!/usr/bin/env r
2 #
3 # A faster lm() replacement based on Armadillo
4 #
5 # This improves on the previous version using GNU GSL
6 #
```

```
7 # Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
8 #
9 # This file is part of Rcpp.
10 #
11 # Rcpp is free software: you can redistribute it and/or modify it
12 # under the terms of the GNU General Public License as published by
13 # the Free Software Foundation, either version 2 of the License, or
14 # (at your option) any later version.
15 #
16 # Rcpp is distributed in the hope that it will be useful, but
17 # WITHOUT ANY WARRANTY; without even the implied warranty of
18 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
19 # GNU General Public License for more details.
20 #
21 # You should have received a copy of the GNU General Public License
22 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
23
24 source("lmArmadillo.R")
25
26 checkLmArmadillo <- function(y, X) {
27     fun <- lmArmadillo()
28     res <- fun(y, X)
29     fit <- lm(y ~ X - 1)
30     rc <- all.equal( as.numeric(res[[1]]), as.numeric(coef(fit))) &
31           all.equal( as.numeric(res[[2]]), as.numeric(coef(summary(fit))[,2]))
32     invisible(rc)
33 }
34
35 timeLmArmadillo <- function(y, X, N) {
36     fun <- lmArmadillo();
37     meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
38 }
39
40 set.seed(42)
41 n <- 5000
42 k <- 9
43 X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
44 truecoef <- 1:(k+1)
45 y <- as.numeric(X %*% truecoef + rnorm(n))
46
47 N <- 100
48
49 stopifnot(checkLmArmadillo(y, X))
50 mt <- timeLmArmadillo(y, X, N)
51 cat("Armadillo: Running", N, "simulations yields (trimmed) mean time", mt, "\n")
```

8.9 FastLM/fastLMviaGSL.r

Runs lm via GSL and times the run

```
1 #!/usr/bin/env r
2 #
3 # A faster lm() replacement based on GNU GSL
4 #
5 # This first appeared in the 'Intro to HPC tutorials'
6 # but has been wrapped in inline::cfunction() here
7 #
8 # Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
9 #
10 # This file is part of Rcpp.
11 #
12 # Rcpp is free software: you can redistribute it and/or modify it
13 # under the terms of the GNU General Public License as published by
14 # the Free Software Foundation, either version 2 of the License, or
15 # (at your option) any later version.
16 #
17 # Rcpp is distributed in the hope that it will be useful, but
18 # WITHOUT ANY WARRANTY; without even the implied warranty of
19 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
20 # GNU General Public License for more details.
21 #
22 # You should have received a copy of the GNU General Public License
23 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
```

```
24
25 source("lmGSL.R")
26
27 checkLmGSL <- function(y, X) {
28     fun <- lmGSL()
29     res <- fun(y, X)
30     fit <- lm(y ~ X - 1)
31     rc <- all.equal( res[[1]], as.numeric(coef(fit))) &
32           all.equal( res[[2]], as.numeric(coef(summary(fit))[,2]))
33     invisible(rc)
34 }
35
36 timeLmGSL <- function(y, X, N) {
37     fun <- lmGSL();
38     meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
39 }
40
41 set.seed(42)
42 n <- 5000
43 k <- 9
44 X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
45 truecoef <- 1:(k+1)
46 y <- as.numeric(X %*% truecoef + rnorm(n))
47
48 N <- 100
49
50 stopifnot(checkLmGSL(y, X))
51 mt <- timeLmGSL(y, X, N)
52 cat("GSL: Running", N, "simulations yields (trimmed) mean time", mt, "\n")
```

8.10 FastLM/lmArmadillo.r

Armadillo-based implementation of a bare-boned lm()

8.11 FastLM/lmGSL.r

GSL-based implementation of a bare-boned lm()

8.12 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.

8.13 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.

8.14 Misc/fibonacci.r

Faster recursive computation of Fibonacci numbers via Rcpp

```
1 #!/usr/bin/env r
2
3 ## this short example was provided in response to this StackOverflow questions:
4 ## http://stackoverflow.com/questions/6807068/why-is-my-recursive-function-so-slow-in-r
5 ## and illustrates that recursive function calls are a) really expensive in R and b) not
6 ## all expensive in C++ (my machine sees a 700-fold speed increase) and c) the byte
7 ## compiler in R does not help here.
8
9 suppressMessages(library(Rcpp))
10
11 ## byte compiler
12 require(compiler)
13
14 ## A C++ version compile with cppFunction
15 fibRcpp <- cppFunction( '
16 int fibonacci(const int x) {
17    if (x == 0) return(0);
18    if (x == 1) return(1);
19    return (fibonacci(x - 1)) + fibonacci(x - 2);
20 }
21 ' )
22
23
24 ## for comparison, the original (but repaired with 0/1 offsets)
25 fibR <- function(seq) {
26     if (seq == 0) return(0);
27     if (seq == 1) return(1);
28     return (fibR(seq - 1) + fibR(seq - 2));
29 }
30
31 ## also use byte-compiled R function
32 fibRC <- cmpfun(fibR)
33
34 ## load rbenchmark to compare
35 library(rbenchmark)
36
37 N <- 35     ## same parameter as original post
38 res <- benchmark(fibR(N),
39                  fibRC(N),
40                  fibRcpp(N),
41                  columns=c("test", "replications", "elapsed",
42                            "relative", "user.self", "sys.self"),
43                  order="relative",
44                  replications=1)
45 print(res)  ## show result
46
```

8.15 Misc/ifelseLooped.r

Nice example of accelerating path-dependent loops with Rcpp

```
1 #!/usr/bin/env r
2 ##
3 ## This example goes back to the following StackOverflow questions:
4 ##
       http://stackoverflow.com/questions/7153586/can-i-vectorize-a-calculation-which-depends-on-previous-elements
5 ## and provides a nice example of how to accelerate path-dependent
6 ## loops which are harder to vectorise.  It lead to the following blog
7 ## post:
8 ##   http://dirk.eddelbuettel.com/blog/2011/08/23#rcpp_for_path_dependent_loops
9 ##
10 ## Thanks to Josh Ulrich for provided a first nice (R-based) answer on
11 ## StackOverflow and for also catching a small oversight in my posted answer.
12 ##
13 ## Dirk Eddelbuettel, 23 Aug 2011
```

```
14 ##
15 ## Copyrighted but of course GPL'ed
16
17
18 library(inline)
19 library(rbenchmark)
20 library(compiler)
21
22 fun1 <- function(z) {
23   for(i in 2:NROW(z)) {
24     z[i] <- ifelse(z[i-1]==1, 1, 0)
25   }
26   z
27 }
28 fun1c <- cmpfun(fun1)
29
30
31 fun2 <- function(z) {
32   for(i in 2:NROW(z)) {
33     z[i] <- if(z[i-1]==1) 1 else 0
34   }
35   z
36 }
37 fun2c <- cmpfun(fun2)
38
39
40 funRcpp <- cxxfunction(signature(zs="numeric"), plugin="Rcpp", body="
41   Rcpp::NumericVector z = Rcpp::NumericVector(zs);
42   int n = z.size();
43   for (int i=1; i<n; i++) {
44     z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
45   }
46   return(z);
47 ")
48
49
50 z <- rep(c(1,1,0,0,0,0), 100)
51 ## test all others against fun1 and make sure all are identical
52 all(sapply(list(fun2(z),fun1c(z),fun2c(z),funRcpp(z)), identical, fun1(z)))
53
54 res <- benchmark(fun1(z), fun2(z),
55                  fun1c(z), fun2c(z),
56                  funRcpp(z),
57                  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
58                  order="relative",
59                  replications=1000)
60 print(res)
61
62 z <- c(1,1,0,0,0,0)
63 res2 <- benchmark(fun1(z), fun2(z),
64                  fun1c(z), fun2c(z),
65                  funRcpp(z),
66                  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
67                  order="relative",
68                  replications=10000)
69 print(res2)
70
71
72 if (FALSE) {                           # quick test to see if Int vectors are faster: appears not
73     funRcppI <- cxxfunction(signature(zs="integer"), plugin="Rcpp", body="
74         Rcpp::IntegerVector z = Rcpp::IntegerVector(zs);
75         int n = z.size();
76         for (int i=1; i<n; i++) {
77             z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
78         }
79         return(z);
80     ")
81
82     z <- rep(c(1L,1L,0L,0L,0L,0L), 100)
83     identical(fun1(z),fun2(z),fun1c(z),fun2c(z),funRcppI(z))
84
85     res3 <- benchmark(fun1(z), fun2(z),
86                       fun1c(z), fun2c(z),
87                       funRcppI(z),
88                       columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
89                       order="relative",
90                       replications=1000)
91     print(res3)
92
93     z <- c(1L,1L,0L,0L,0L,0L)
94     res4 <- benchmark(fun1(z), fun2(z),
```

```
95                      fun1c(z), fun2c(z),
96                      funRcppI(z),
97                      columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
98                      order="relative",
99                      replications=10000)
100     print(res4)
101 }
```

8.16 OpenMP/check.R

Compares both approaches

```
1
2 dyn.load("piWithInterrupts.so")
3 #res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
4 res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
5 print(res, digits=10)
```

8.17 OpenMP/OpenMPandInline.r

Example of using OpenMP compilation via inline

```
1 #!/usr/bin/env r
2
3 library(inline)
4 library(rbenchmark)
5
6 serialCode <- '
7    // assign to C++ vector
8    std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
9    size_t n = x.size();
10    for (size_t i=0; i<n; i++) {
11        x[i] = ::log(x[i]);
12    }
13    return Rcpp::wrap(x);
14 '
15 funSerial <- cxxfunction(signature(xs="numeric"), body=serialCode, plugin="Rcpp")
16
17 serialStdAlgCode <- '
18    std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
19    std::transform(x.begin(), x.end(), x.begin(), ::log);
20    return Rcpp::wrap(x);
21 '
22 funSerialStdAlg <- cxxfunction(signature(xs="numeric"), body=serialStdAlgCode, plugin="Rcpp")
23
24 ## same, but with Rcpp vector just to see if there is measurable difference
25 serialRcppCode <- '
26    // assign to C++ vector
27    Rcpp::NumericVector x = Rcpp::NumericVector(xs);
28    size_t n = x.size();
29    for (size_t i=0; i<n; i++) {
30        x[i] = ::log(x[i]);
31    }
32    return x;
33 '
34 funSerialRcpp <- cxxfunction(signature(xs="numeric"), body=serialRcppCode, plugin="Rcpp")
35
36 serialStdAlgRcppCode <- '
37    Rcpp::NumericVector x = Rcpp::NumericVector(xs);
38    std::transform(x.begin(), x.end(), x.begin(), ::log);
39    return x;
40 '
41 funSerialStdAlgRcpp <- cxxfunction(signature(xs="numeric"), body=serialStdAlgRcppCode, plugin="Rcpp")
42
43 serialImportTransRcppCode <- '
44    Rcpp::NumericVector x(xs);
```

```
45    return Rcpp::NumericVector::import_transform(x.begin(), x.end(), ::log);
46 '
47 funSerialImportTransRcpp <- cxxfunction(signature(xs="numeric"), body=serialImportTransRcppCode,
       plugin="Rcpp")
48
49 ## now with a sugar expression with internalizes the loop
50 sugarRcppCode <- '
51    // assign to C++ vector
52    Rcpp::NumericVector x = log ( Rcpp::NumericVector(xs) );
53    return x;
54 '
55 funSugarRcpp <- cxxfunction(signature(xs="numeric"), body=sugarRcppCode, plugin="Rcpp")
56
57 ## lastly via OpenMP for parallel use
58 openMPCode <- '
59    // assign to C++ vector
60    std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
61    size_t n = x.size();
62 #pragma omp parallel for shared(x, n)
63    for (size_t i=0; i<n; i++) {
64        x[i] = ::log(x[i]);
65    }
66    return Rcpp::wrap(x);
67 '
68
69 ## modify the plugin for Rcpp to support OpenMP
70 settings <- getPlugin("Rcpp")
71 settings$env$PKG_CXXFLAGS <- paste('-fopenmp', settings$env$PKG_CXXFLAGS)
72 settings$env$PKG_LIBS <- paste('-fopenmp -lgomp', settings$env$PKG_LIBS)
73
74 funOpenMP <- cxxfunction(signature(xs="numeric"), body=openMPCode, plugin="Rcpp", settings=settings)
75
76
77 z <- seq(1, 2e6)
78 res <- benchmark(funSerial(z), funSerialStdAlg(z),
79                  funSerialRcpp(z), funSerialStdAlgRcpp(z),
80                  funSerialImportTransRcpp(z),
81                  funOpenMP(z), funSugarRcpp(z),
82                  columns=c("test", "replications", "elapsed",
83                            "relative", "user.self", "sys.self"),
84                  order="relative",
85                  replications=100)
86 print(res)
87
88
```

8.18 OpenMP/piWithInterrupts.cpp.R

Computing pi via MPI, and letting the user interupt via Ctrl-C

8.19 RcppGibbs/RcppGibbs.R

Implementation of a MCMC Gibbs sampler using Rcpp

```
1 ## Simple Gibbs Sampler Example
2 ## Adapted from Darren Wilkinson's post at
3 ## http://darrenjw.wordpress.com/2010/04/28/mcmc-programming-in-r-python-java-and-c/
4 ##
5 ## Sanjog Misra and Dirk Eddelbuettel, June-July 2011
6
7 suppressMessages(library(Rcpp))
8 suppressMessages(library(inline))
9 suppressMessages(library(compiler))
10 suppressMessages(library(rbenchmark))
```

```
11
12
13 ## Actual joint density -- the code which follow implements
14 ## a Gibbs sampler to draw from the following joint density f(x,y)
15 fun <- function(x,y) {
16     x*x * exp(-x*y*y - y*y + 2*y - 4*x)
17 }
18
19 ## Note that the full conditionals are propotional to
20 ## f(x|y) = (x^2)*exp(-x*(4+y*y))              : a Gamma density kernel
21 ## f(y|x) = exp(-0.5*2*(x+1)*(y^2 - 2*y/(x+1)) : Normal Kernel
22
23 ## There is a small typo in Darrens code.
24 ## The full conditional for the normal has the wrong variance
25 ## It should be 1/sqrt(2*(x+1)) not 1/sqrt(1+x)
26 ## This we can verify ...
27 ## The actual conditional (say for x=3) can be computed as follows
28 ## First - Construct the Unnormalized Conditional
29 fy.unnorm <- function(y) fun(3,y)
30
31 ## Then - Find the appropriate Normalizing Constant
32 K <- integrate(fy.unnorm,-Inf,Inf)
33
34 ## Finally - Construct Actual Conditional
35 fy <- function(y) fy.unnorm(y)/K$val
36
37 ## Now - The corresponding Normal should be
38 fy.dnorm <- function(y) {
39     x <- 3
40     dnorm(y,1/(1+x),sqrt(1/(2*(1+x))))
41 }
42
43 ## and not ...
44 fy.dnorm.wrong <- function(y) {
45     x <- 3
46     dnorm(y,1/(1+x),sqrt(1/((1+x))))
47 }
48
49 if (interactive()) {
50     ## Graphical check
51     ## Actual (gray thick line)
52     curve(fy,-2,2,col='grey',lwd=5)
53
54     ## Correct Normal conditional (blue dotted line)
55     curve(fy.dnorm,-2,2,col='blue',add=T,lty=3)
56
57     ## Wrong Normal (Red line)
58     curve(fy.dnorm.wrong,-2,2,col='red',add=T)
59 }
60
61 ## Here is the actual Gibbs Sampler
62 ## This is Darren Wilkinsons R code (with the corrected variance)
63 ## But we are returning only his columns 2 and 3 as the 1:N sequence
64 ## is never used below
65 Rgibbs <- function(N,thin) {
66     mat <- matrix(0,ncol=2,nrow=N)
67     x <- 0
68     y <- 0
69     for (i in 1:N) {
70         for (j in 1:thin) {
71             x <- rgamma(1,3,y*y+4)
72             y <- rnorm(1,1/(x+1),1/sqrt(2*(x+1)))
73         }
74         mat[i,] <- c(x,y)
75     }
76     mat
77 }
78
79 ## We can also try the R compiler on this R function
80 RCgibbs <- cmpfun(Rgibbs)
81
82 ## For example
83 ## mat <- Rgibbs(10000,10); dim(mat)
84 ## would give: [1] 10000     2
85
86 ## Now for the Rcpp version -- Notice how easy it is to code up!
87
88 ## NOTE: This is the old way to compile Rcpp code inline.
89 ## The code here has left as a historical artifact and tribute to the old way.
90 ## Please use the code under the "new" inline compilation section.
91
```

```
92 gibbscode <- '
93
94   using namespace Rcpp;   // inline does that for us already
95
96   // n and thin are SEXPs which the Rcpp::as function maps to C++ vars
97   int N   = as<int>(n);
98   int thn = as<int>(thin);
99
100   int i,j;
101   NumericMatrix mat(N, 2);
102
103   RNGScope scope;         // Initialize Random number generator
104
105   // The rest of the code follows the R version
106   double x=0, y=0;
107
108   for (i=0; i<N; i++) {
109     for (j=0; j<thn; j++) {
110       x = ::Rf_rgamma(3.0,1.0/(y*y+4));
111       y = ::Rf_rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
112     }
113     mat(i,0) = x;
114     mat(i,1) = y;
115   }
116
117   return mat;             // Return to R
118 '
119
120 # Compile and Load
121 RcppGibbs_old <- cxxfunction(signature(n="int", thin = "int"),
122                          gibbscode, plugin="Rcpp")
123
124
125 gslgibbsincl <- '
126   #include <gsl/gsl_rng.h>
127   #include <gsl/gsl_randist.h>
128
129   using namespace Rcpp;  // just to be explicit
130 '
131
132 gslgibbscode <- '
133   int N = as<int>(ns);
134   int thin = as<int>(thns);
135   int i, j;
136   gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
137   double x=0, y=0;
138   NumericMatrix mat(N, 2);
139   for (i=0; i<N; i++) {
140     for (j=0; j<thin; j++) {
141       x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
142       y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
143     }
144     mat(i,0) = x;
145     mat(i,1) = y;
146   }
147   gsl_rng_free(r);
148
149   return mat;           // Return to R
150 '
151
152 ## Compile and Load
153 GSLGibbs_old <- cxxfunction(signature(ns="int", thns = "int"),
154                          body=gslgibbscode, includes=gslgibbsincl,
155                          plugin="RcppGSL")
156
157 ## without RcppGSL, using cfunction()
158 #GSLGibbs <- cfunction(signature(ns="int", thns = "int"),
159 #                      body=gslgibbscode, includes=gslgibbsincl,
160 #                      Rcpp=TRUE,
161 #                      cppargs="-I/usr/include",
162 #                      libargs="-lgsl -lgslcblas")
163
164
165 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
166 ## written. Please use the code next as a template for your own project.
167
168 ## Use of the cppFunction() gives the ability to immediately compile embed C++
169 ## without having to worry about header specification or Rcpp attributes.
170
171 cppFunction('
172 NumericMatrix RcppGibbs(int N, int thn){
```

```
173     // Note: n and thin are SEXPs which the Rcpp automatically converts to ints
174
175     // Setup storage matrix
176     NumericMatrix mat(N, 2);
177
178     // The rest of the code follows the R version
179     double x = 0, y = 0;
180
181     for (int i = 0; i < N; i++) {
182         for (int j = 0; j < thn; j++) {
183             x = R::rgamma(3.0,1.0/(y*y+4));
184             y = R::rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
185         }
186         mat(i,0) = x;
187         mat(i,1) = y;
188     }
189
190     return mat;             // Return to R
191 }')
192
193
194 ## Use of the sourceCpp() is preferred for users who wish to source external
195 ## files or specify their headers and Rcpp attributes within their code.
196 ## Code here is able to easily be extracted and placed into its own C++ file.
197
198 ## Compile and Load
199 sourceCpp(code="
200 #include <RcppGSL.h>
201 #include <gsl/gsl_rng.h>
202 #include <gsl/gsl_randist.h>
203
204 using namespace Rcpp;  // just to be explicit
205
206 // [[Rcpp::depends(RcppGSL)]]
207
208 // [[Rcpp::export]]
209 NumericMatrix GSLGibbs(int N, int thin){
210     gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
211     double x = 0, y = 0;
212     NumericMatrix mat(N, 2);
213     for (int i = 0; i < N; i++) {
214         for (int j = 0; j < thin; j++) {
215             x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
216             y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
217         }
218         mat(i,0) = x;
219         mat(i,1) = y;
220     }
221     gsl_rng_free(r);
222
223     return mat;           // Return to R
224 }")
225
226
227
228 ## Now for some tests
229 ## You can try other values if you like
230 ## Note that the total number of interations are N*thin!
231 Ns <- c(1000,5000,10000,20000)
232 thins <- c(10,50,100,200)
233 tim_R <- rep(0,4)
234 tim_RC <- rep(0,4)
235 tim_Rgsl <- rep(0,4)
236 tim_Rcpp <- rep(0,4)
237
238 for (i in seq_along(Ns)) {
239     tim_R[i] <- system.time(mat <- Rgibbs(Ns[i],thins[i]))[3]
240     tim_RC[i] <- system.time(cmat <- RCgibbs(Ns[i],thins[i]))[3]
241     tim_Rgsl[i] <- system.time(gslmat <- GSLGibbs(Ns[i],thins[i]))[3]
242     tim_Rcpp[i] <- system.time(rcppmat <- RcppGibbs(Ns[i],thins[i]))[3]
243     cat("Replication #", i, "complete \n")
244 }
245
246 ## Comparison
247 speedup <- round(tim_R/tim_Rcpp,2);
248 speedup2 <- round(tim_R/tim_Rgsl,2);
249 speedup3 <- round(tim_R/tim_RC,2);
250 summtab <- round(rbind(tim_R,tim_RC, tim_Rcpp,tim_Rgsl,speedup3,speedup,speedup2),3)
251 colnames(summtab) <- c("N=1000","N=5000","N=10000","N=20000")
252 rownames(summtab) <- c("Elasped Time (R)","Elasped Time (RC)","Elapsed Time (Rcpp)", "Elapsed Time (Rgsl)",
253                        "SpeedUp Rcomp.","SpeedUp Rcpp", "SpeedUp GSL")
```

```
254
255 print(summtab)
256
257 ## Contour Plots -- based on Darren's example
258 if (interactive() && require(KernSmooth)) {
259     op <- par(mfrow=c(4,1),mar=c(3,3,3,1))
260     x <- seq(0,4,0.01)
261     y <- seq(-2,4,0.01)
262     z <- outer(x,y,fun)
263     contour(x,y,z,main="Contours of actual distribution",xlim=c(0,2), ylim=c(-2,4))
264     fit <- bkde2D(as.matrix(mat),c(0.1,0.1))
265     contour(drawlabels=T, fit$x1, fit$x2, fit$fhat, xlim=c(0,2), ylim=c(-2,4),
266             main=paste("Contours of empirical distribution:",round(tim_R[4],2)," seconds"))
267     fitc <- bkde2D(as.matrix(rcppmat),c(0.1,0.1))
268     contour(fitc$x1,fitc$x2,fitc$fhat,xlim=c(0,2), ylim=c(-2,4),
269             main=paste("Contours of Rcpp based empirical distribution:",round(tim_Rcpp[4],2)," seconds"))
270     fitg <- bkde2D(as.matrix(gslmat),c(0.1,0.1))
271     contour(fitg$x1,fitg$x2,fitg$fhat,xlim=c(0,2), ylim=c(-2,4),
272             main=paste("Contours of GSL based empirical distribution:",round(tim_Rgsl[4],2)," seconds"))
273     par(op)
274 }
275
276
277 ## also use rbenchmark package
278 N <- 20000
279 thn <- 200
280 res <- benchmark(Rgibbs(N, thn),
281                  RCgibbs(N, thn),
282                  RcppGibbs(N, thn),
283                  GSLGibbs(N, thn),
284                  columns=c("test", "replications", "elapsed",
285                            "relative", "user.self", "sys.self"),
286                  order="relative",
287                  replications=10)
288 print(res)
289
290
291 ## And we are done
292
293
```

8.20 RcppGibbs/timeRNGs.R

Illustration of speed differences between Nornal and Uniform RNG draws from R and GSL

```
1
2 suppressMessages(library(Rcpp))
3 suppressMessages(library(inline))
4 suppressMessages(library(rbenchmark))
5
6 ## NOTE: This is the old way to compile Rcpp code inline.
7 ## The code here has left as a historical artifact and tribute to the old way.
8 ## Please use the code under the "new" inline compilation section.
9
10 rcppGamma_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body='
11   NumericVector x(xs);
12   int n   = x.size();
13
14   // Initialize Random number generator
15   RNGScope scope;
16
17   const double y = 1.234;
18   for (int i=0; i<n; i++) {
19     x[i] = ::Rf_rgamma(3.0, 1.0/(y*y+4));
20   }
21
22   // Return to R
23   return x;
24 ')
25
26
27 gslGamma_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
28                         include='#include <gsl/gsl_rng.h>
```

```
29                                  #include <gsl/gsl_randist.h>',
30                          body='
31   NumericVector x(xs);
32   int n   = x.size();
33
34   gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
35   const double y = 1.234;
36   for (int i=0; i<n; i++) {
37     x[i] = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
38   }
39   gsl_rng_free(r);
40
41   // Return to R
42   return x;
43 ')
44
45
46 rcppNormal_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body='
47   NumericVector x(xs);
48   int n   = x.size();
49
50   // Initialize Random number generator
51   RNGScope scope;
52
53   const double y = 1.234;
54   for (int i=0; i<n; i++) {
55     x[i] = ::Rf_rnorm(1.0/(y+1),1.0/sqrt(2*y+2));
56   }
57
58   // Return to R
59   return x;
60 ')
61
62
63 gslNormal_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
64                          include='#include <gsl/gsl_rng.h>
65                                  #include <gsl/gsl_randist.h>',
66                          body='
67   NumericVector x(xs);
68   int n   = x.size();
69
70   gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
71   const double y = 1.234;
72   for (int i=0; i<n; i++) {
73     x[i] = 1.0/(y+1)+gsl_ran_gaussian(r,1.0/sqrt(2*y+2));
74   }
75   gsl_rng_free(r);
76
77   // Return to R
78   return x;
79 ')
80
81
82 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
83 ## written. Please use the code next as a template for your own project.
84
85 cppFunction('
86 NumericVector rcppGamma(NumericVector x){
87     int n   = x.size();
88
89     const double y = 1.234;
90     for (int i=0; i<n; i++) {
91         x[i] = R::rgamma(3.0, 1.0/(y*y+4));
92     }
93
94     // Return to R
95     return x;
96 }')
97
98 ## This approach is a bit sloppy. Generally, you will want to use
99 ## sourceCpp() if there are additional includes that are required.
100 cppFunction('
101 NumericVector gslGamma(NumericVector x){
102     int n   = x.size();
103
104     gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
105     const double y = 1.234;
106     for (int i=0; i<n; i++) {
107         x[i] = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
108     }
109     gsl_rng_free(r);
```

```
110
111     // Return to R
112     return x;
113 }', includes = '#include <gsl/gsl_rng.h>
114                  #include <gsl/gsl_randist.h>',
115     depends = "RcppGSL")
116
117
118 cppFunction('
119 NumericVector rcppNormal(NumericVector x){
120     int n   = x.size();
121
122     const double y = 1.234;
123     for (int i=0; i<n; i++) {
124         x[i] = R::rnorm(1.0/(y+1),1.0/sqrt(2*y+2));
125     }
126
127     // Return to R
128     return x;
129 }')
130
131
132 ## Here we demonstrate the use of sourceCpp() to show the continuity
133 ## of the code artifact.
134
135 sourceCpp(code = '
136 #include <RcppGSL.h>
137 #include <gsl/gsl_rng.h>
138 #include <gsl/gsl_randist.h>
139
140 using namespace Rcpp;
141
142 // [[Rcpp::depends("RcppGSL")]]
143
144 // [[Rcpp::export]]
145 NumericVector gslNormal(NumericVector x){
146     int n   = x.size();
147
148     gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
149     const double y = 1.234;
150     for (int i=0; i<n; i++) {
151         x[i] = 1.0/(y+1)+gsl_ran_gaussian(r,1.0/sqrt(2*y+2));
152     }
153     gsl_rng_free(r);
154
155     // Return to R
156     return x;
157 }')
158
159 x <- rep(NA, 1e6)
160 res <- benchmark(rcppGamma(x),
161                  gslGamma(x),
162                  rcppNormal(x),
163                  gslNormal(x),
164                  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
165                  order="relative",
166                  replications=20)
167 print(res)
168
169
```

8.21 RcppInline/external_pointer.r

A simple example (using inline) of external Pointer use via Rcpp::XPtr

```
1 #!/usr/bin/env r
2 #
3 # Copyright (C) 2009 - 2010     Romain Francois
4 #
5 # This file is part of Rcpp.
6 #
7 # Rcpp is free software: you can redistribute it and/or modify it
8 # under the terms of the GNU General Public License as published by
```

```
9 # the Free Software Foundation, either version 2 of the License, or
10 # (at your option) any later version.
11 #
12 # Rcpp is distributed in the hope that it will be useful, but
13 # WITHOUT ANY WARRANTY; without even the implied warranty of
14 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
15 # GNU General Public License for more details.
16 #
17 # You should have received a copy of the GNU General Public License
18 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
19
20 require(Rcpp)
21 require(inline)
22
23
24 ## NOTE: This is the old way to compile Rcpp code inline.
25 ## The code here has left as a historical artifact and tribute to the old way.
26 ## Please use the code under the "new" inline compilation section.
27
28 funx_old <- cxxfunction(signature(), '
29         /* creating a pointer to a vector<int> */
30         std::vector<int>* v = new std::vector<int> ;
31         v->push_back( 1 ) ;
32         v->push_back( 2 ) ;
33
34         /* wrap the pointer as an external pointer */
35         /* this automatically protected the external pointer from R garbage
36            collection until p goes out of scope. */
37         Rcpp::XPtr< std::vector<int> > p(v) ;
38
39         /* return it back to R, since p goes out of scope after the return
40            the external pointer is no more protected by p, but it gets
41            protected by being on the R side */
42         return( p ) ;
43 ', plugin = "Rcpp" )
44 xp <- funx_old()
45 stopifnot( identical( typeof( xp ), "externalptr" ) )
46
47 # passing the pointer back to C++
48 funx_old <- cxxfunction(signature(x = "externalptr" ), '
49         /* wrapping x as smart external pointer */
50         /* The SEXP based constructor does not protect the SEXP from
51            garbage collection automatically, it is already protected
52            because it comes from the R side, however if you want to keep
53            the Rcpp::XPtr object on the C(++) side
54            and return something else to R, you need to protect the external
55            pointer, by using the protect member function */
56         Rcpp::XPtr< std::vector<int> > p(x) ;
57
58         /* just return the front of the vector as a SEXP */
59         return( Rcpp::wrap( p->front() ) ) ;
60 ', plugin = "Rcpp" )
61 front <- funx_old(xp)
62 stopifnot( identical( front, 1L ) )
63
64
65 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
66 ## written. Please use the code next as a template for your own project.
67
68 ## Use of the cppFunction() gives the ability to immediately compile embedded
69 ## C++ directly within R without having to worry about header specification or
70 ## Rcpp attributes.
71
72 cppFunction('
73 Rcpp::XPtr< std::vector<int> >  funx(){
74     /* creating a pointer to a vector<int> */
75     std::vector<int>* v = new std::vector<int> ;
76     v->push_back( 1 ) ;
77     v->push_back( 2 ) ;
78
79     /* wrap the pointer as an external pointer */
80     /* this automatically protected the external pointer from R garbage
81      *   collection until p goes out of scope.
82      */
83     Rcpp::XPtr< std::vector<int> > p(v) ;
84
85     /* return it back to R, since p goes out of scope after the return
86      * the external pointer is no more protected by p, but it gets
87      * protected by being on the R side
88      */
89     return( p ) ;
```

```
90 }')
91
92 xp <- funx()
93 stopifnot( identical( typeof( xp ), "externalptr" ) )
94
95 # passing the pointer back to C++
96 cppFunction('
97 SEXP funx_pt(Rcpp::XPtr< std::vector<int> > p){
98     /* Wrapping x as smart external pointer */
99
100     /* The SEXP based constructor does not protect the SEXP from
101      * garbage collection automatically, it is already protected
102      * because it comes from the R side, however if you want to keep
103      * the Rcpp::XPtr object on the C(++) side
104      * and return something else to R, you need to protect the external
105      * pointer, by using the protect member function
106      */
107
108     /* Just return the front of the vector as a SEXP */
109     return Rcpp::wrap(p->front());
110 }')
111 front <- funx_pt(xp)
112 stopifnot( identical( front, 1L ) )
113
```

8.22 RcppInline/RcppInlineExample.r

The 'distribution of determinant' example as a demonstration of how to use inline and Rcpp for a simple function.

```
1 #!/usr/bin/env r
2
3 suppressMessages(library(Rcpp))
4
5 ## NOTE: This is the old way to compile Rcpp code inline.
6 ## The code here has left as a historical artifact and tribute to the old way.
7 ## Please use the code under the "new" inline compilation section.
8
9 suppressMessages(library(inline))
10
11 foo <- '
12     IntegerVector vec(10000);     // vec parameter viewed as vector of ints.
13     int i = 0;
14     for (int a = 0; a < 9; a++)
15       for (int b = 0; b < 9; b++)
16         for (int c = 0; c < 9; c++)
17           for (int d = 0; d < 9; d++)
18             vec(i++) = a*b - c*d;
19
20     return vec;
21 '
22
23 funx_old <- cxxfunction(signature(), foo, plugin = "Rcpp" )
24
25 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
26 ## written. Please use the code next as a template for your own project.
27
28 cppFunction('IntegerVector funx(){
29     IntegerVector vec(10000);     // vec parameter viewed as vector of ints.
30     int i = 0;
31     for (int a = 0; a < 9; a++)
32       for (int b = 0; b < 9; b++)
33         for (int c = 0; c < 9; c++)
34           for (int d = 0; d < 9; d++)
35             vec(i++) = a*b - c*d;
36
37     return vec;
38 }')
39
40 dd.inline.rcpp <- function() {
41     res <- funx()
42     tabulate(res)
43 }
44
45 print(mean(replicate(100,system.time(dd.inline.rcpp())["elapsed"]),trim=0.05))
46
47
```

8.23 RcppInline/RcppInlineWithLibsExamples.r

Four simple examples of how to combine Rcpp and an external library (where we use the GNU GSL) using inline.

```
1 #!/usr/bin/env r
2 #
3 # Copyright (C) 2009 - 2016  Dirk Eddelbuettel and Romain Francois
4 #
5 # This file is part of Rcpp.
6 #
7 # Rcpp is free software: you can redistribute it and/or modify it
8 # under the terms of the GNU General Public License as published by
9 # the Free Software Foundation, either version 2 of the License, or
10 # (at your option) any later version.
11 #
12 # Rcpp is distributed in the hope that it will be useful, but
13 # WITHOUT ANY WARRANTY; without even the implied warranty of
14 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
15 # GNU General Public License for more details.
16 #
17 # You should have received a copy of the GNU General Public License
18 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
19
20 suppressMessages(library(Rcpp))
21 suppressMessages(library(RcppGSL))
22
23 ## NOTE: This is the old way to compile Rcpp code inline.
24 ## The code here has left as a historical artifact and tribute to the old way.
25 ## Please use the code under the "new" inline compilation section.
26
27 suppressMessages(library(inline))
28
29 firstExample_old <- function() {
30     ## a really simple C program calling three functions from the GSL
31     gslrng <- '
32     gsl_rng *r;
33     gsl_rng_env_setup();
34     double v;
35
36     r = gsl_rng_alloc (gsl_rng_default);
37
38     printf("  generator type: %s\\n", gsl_rng_name (r));
39     printf("  seed = %lu\\n", gsl_rng_default_seed);
40     v = gsl_rng_get (r);
41     printf("  first value = %.0f\\n", v);
42
43     gsl_rng_free(r);
44     return R_NilValue;
45     '
46
47     ## turn into a function that R can call
48     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
49     funx_old <- cxxfunction(signature(), gslrng,
50                             includes="#include <gsl/gsl_rng.h>",
51                             plugin="RcppGSL")
52
53     cat("Calling first example\n")
54     funx_old()
55     invisible(NULL)
56 }
57
58 secondExample_old <- function() {
59
60     ## now use Rcpp to pass down a parameter for the seed
61     gslrng <- '
62     int seed = Rcpp::as<int>(par) ;
63
64     gsl_rng *r;
65     gsl_rng_env_setup();
66     double v;
67
68     r = gsl_rng_alloc (gsl_rng_default);
69
70     gsl_rng_set (r, (unsigned long) seed);
71     v = gsl_rng_get (r);
72
73     #ifndef BeSilent
```

```
74      printf("  generator type: %s\\n", gsl_rng_name (r));
75      printf("  seed = %d\\n", seed);
76      printf("  first value = %.0f\\n", v);
77      #endif
78
79      gsl_rng_free(r);
80      return Rcpp::wrap(v) ;
81      '
82
83      ## turn into a function that R can call
84      ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
85      ## use additional define for compile to suppress output
86      funx_old <- cxxfunction(signature(par="numeric"), gslrng,
87                              includes="#include <gsl/gsl_rng.h>",
88                              plugin="RcppGSL")
89      cat("\n\nCalling second example without -DBeSilent set\n")
90      print(funx_old(0))
91
92
93      ## now override settings to add -D flag
94      settings <- getPlugin("RcppGSL")
95      settings$env$PKG_CPPFLAGS <- paste(settings$PKG_CPPFLAGS, "-DBeSilent")
96
97      funx_old <- cxxfunction(signature(par="numeric"), gslrng,
98                              includes="#include <gsl/gsl_rng.h>",
99                              settings=settings)
100      cat("\n\nCalling second example with -DBeSilent set\n")
101      print(funx_old(0))
102
103     invisible(NULL)
104 }
105
106 thirdExample_old <- function() {
107
108     ## now use Rcpp to pass down a parameter for the seed, and a vector size
109     gslrng <- '
110     int seed = Rcpp::as<int>(s) ;
111     int len = Rcpp::as<int>(n);
112
113     gsl_rng *r;
114     gsl_rng_env_setup();
115     std::vector<double> v(len);
116
117     r = gsl_rng_alloc (gsl_rng_default);
118
119     gsl_rng_set (r, (unsigned long) seed);
120     for (int i=0; i<len; i++) {
121        v[i] = gsl_rng_get (r);
122     }
123     gsl_rng_free(r);
124
125     return Rcpp::wrap(v) ;
126     '
127
128     ## turn into a function that R can call
129     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
130     ## use additional define for compile to suppress output
131     funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
132                             gslrng,
133                             includes="#include <gsl/gsl_rng.h>",
134                             plugin="RcppGSL")
135     cat("\n\nCalling third example with seed and length\n")
136     print(funx_old(0, 5))
137
138     invisible(NULL)
139 }
140
141 fourthExample_old <- function() {
142
143     ## now use Rcpp to pass down a parameter for the seed, and a vector size
144     gslrng <- '
145     int seed = Rcpp::as<int>(s);
146     int len = Rcpp::as<int>(n);
147
148     gsl_rng *r;
149     gsl_rng_env_setup();
150     std::vector<double> v(len);
151
152     r = gsl_rng_alloc (gsl_rng_default);
153
154     gsl_rng_set (r, (unsigned long) seed);
```

```
155     for (int i=0; i<len; i++) {
156        v[i] = gsl_rng_get (r);
157     }
158     gsl_rng_free(r);
159 
160     return wrap(v);
161     '
162 
163     ## turn into a function that R can call
164     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
165     ## use additional define for compile to suppress output
166     funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
167                             gslrng,
168                             includes=c("#include <gsl/gsl_rng.h>",
169                                        "using namespace Rcpp;",
170                                        "using namespace std;"),
171                             plugin="RcppGSL")
172     cat("\n\nCalling fourth example with seed, length and namespaces\n")
173     print(funx_old(0, 5))
174 
175     invisible(NULL)
176 }
177 
178 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
179 ## written. Please use the code next as a template for your own project.
180 
181 firstExample <- function() {
182     ## a really simple C program calling three functions from the GSL
183 
184     sourceCpp(code='
185 #include <RcppGSL.h>
186 #include <gsl/gsl_rng.h>
187 
188 // [[Rcpp::depends(RcppGSL)]]
189 
190 // [[Rcpp::export]]
191 SEXP funx(){
192     gsl_rng *r;
193     gsl_rng_env_setup();
194     double v;
195 
196     r = gsl_rng_alloc (gsl_rng_default);
197 
198     printf("  generator type: %s\\n", gsl_rng_name (r));
199     printf("  seed = %lu\\n", gsl_rng_default_seed);
200     v = gsl_rng_get (r);
201     printf("  first value = %.0f\\n", v);
202 
203     gsl_rng_free(r);
204     return R_NilValue;
205 }')
206 
207     cat("Calling first example\n")
208     funx()
209     invisible(NULL)
210 }
211 
212 secondExample <- function() {
213 
214     ## now use Rcpp to pass down a parameter for the seed
215 
216     ## turn into a function that R can call
217     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
218     ## use additional define for compile to suppress output
219 
220     gslrng <- '
221     #include <RcppGSL.h>
222     #include <gsl/gsl_rng.h>
223 
224     // [[Rcpp::depends(RcppGSL)]]
225 
226     // [[Rcpp::export]]
227     double funx(int seed){
228 
229     gsl_rng *r;
230     gsl_rng_env_setup();
231     double v;
232 
233     r = gsl_rng_alloc (gsl_rng_default);
234 
235     gsl_rng_set (r, (unsigned long) seed);
```

```
236     v = gsl_rng_get (r);
237
238     #ifndef BeSilent
239     printf("  generator type: %s\\n", gsl_rng_name (r));
240     printf("  seed = %d\\n", seed);
241     printf("  first value = %.0f\\n", v);
242     #endif
243
244     gsl_rng_free(r);
245     return v;
246     }'
247
248     sourceCpp(code=gslrng, rebuild = TRUE)
249
250     cat("\n\nCalling second example without -DBeSilent set\n")
251     print(funx(0))
252
253
254     ## now override settings to add -D flag
255     o = Sys.getenv("PKG_CPPFLAGS")
256     Sys.setenv("PKG_CPPFLAGS" = paste(o, "-DBeSilent"))
257
258     sourceCpp(code=gslrng, rebuild = TRUE)
259
260     # Restore environment flags
261     Sys.setenv("PKG_CPPFLAGS" = o )
262
263     cat("\n\nCalling second example with -DBeSilent set\n")
264     print(funx(0))
265
266     invisible(NULL)
267 }
268
269 thirdExample <- function() {
270
271     ## now use Rcpp to pass down a parameter for the seed, and a vector size
272
273     ## turn into a function that R can call
274     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
275     ## use additional define for compile to suppress output
276
277     sourceCpp(code='
278     #include <RcppGSL.h>
279     #include <gsl/gsl_rng.h>
280
281     // [[Rcpp::depends(RcppGSL)]]
282
283     // [[Rcpp::export]]
284     std::vector<double> funx(int seed, int len){
285
286     gsl_rng *r;
287     gsl_rng_env_setup();
288     std::vector<double> v(len);
289
290     r = gsl_rng_alloc (gsl_rng_default);
291
292     gsl_rng_set (r, (unsigned long) seed);
293     for (int i=0; i<len; i++) {
294        v[i] = gsl_rng_get (r);
295     }
296     gsl_rng_free(r);
297
298     return v;
299     }')
300
301     cat("\n\nCalling third example with seed and length\n")
302     print(funx(0, 5))
303
304     invisible(NULL)
305 }
306
307 fourthExample <- function() {
308
309     ## now use Rcpp to pass down a parameter for the seed, and a vector size
310
311     ## turn into a function that R can call
312     ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
313     ## use additional define for compile to suppress output
314
315     sourceCpp(code='
316     #include <RcppGSL.h>
```

```
317     #include <gsl/gsl_rng.h>
318
319     using namespace Rcpp;
320     using namespace std;
321
322     // [[Rcpp::depends(RcppGSL)]]
323
324     // [[Rcpp::export]]
325     std::vector<double> funx(int seed, int len){
326
327     gsl_rng *r;
328     gsl_rng_env_setup();
329     std::vector<double> v(len);
330
331     r = gsl_rng_alloc (gsl_rng_default);
332
333     gsl_rng_set (r, (unsigned long) seed);
334     for (int i=0; i<len; i++) {
335     v[i] = gsl_rng_get (r);
336     }
337     gsl_rng_free(r);
338
339     return v;
340     }')
341
342     cat("\n\nCalling fourth example with seed, length and namespaces\n")
343     print(funx(0, 5))
344
345     invisible(NULL)
346 }
347
348 firstExample()
349 secondExample()
350 thirdExample()
351 fourthExample()
```

8.24 RcppInline/RcppSimpleExample.r

A very simple example of using inline without Rcpp.

```
1 #!/usr/bin/env r
2
3
4 suppressMessages(library(Rcpp))
5 suppressMessages(library(inline))
6
7
8 foo <- '
9   int i, j, na, nb, nab;
10   double *xa, *xb, *xab;
11   SEXP ab;
12
13   PROTECT(a = AS_NUMERIC(a));
14   PROTECT(b = AS_NUMERIC(b));
15   na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
16   PROTECT(ab = NEW_NUMERIC(nab));
17   xa = NUMERIC_POINTER(a); xb = NUMERIC_POINTER(b);
18   xab = NUMERIC_POINTER(ab);
19   for(i = 0; i < nab; i++) xab[i] = 0.0;
20   for(i = 0; i < na; i++)
21     for(j = 0; j < nb; j++) xab[i + j] += xa[i] * xb[j];
22   UNPROTECT(3);
23   return(ab);
24 '
25
26 funx <- cfunction(signature(a="numeric",b="numeric"), foo, Rcpp=FALSE, verbose=FALSE)
27 funx(a=1:20, b=2:11)
```

8.25 RcppInline/RcppSimpleTests.r

This file provides a few simple tests that preceded the creation of the numerous formal unit tests that followed.

8.26 RcppInline/RObject.r

A number of examples on how to use wrap() to automatically convert data types.

```
1 #!/usr/bin/env r
2 #
3 # Copyright (C) 2009 - 2010  Dirk Eddelbuettel and Romain Francois
4 #
5 # This file is part of Rcpp.
6 #
7 # Rcpp is free software: you can redistribute it and/or modify it
8 # under the terms of the GNU General Public License as published by
9 # the Free Software Foundation, either version 2 of the License, or
10 # (at your option) any later version.
11 #
12 # Rcpp is distributed in the hope that it will be useful, but
13 # WITHOUT ANY WARRANTY; without even the implied warranty of
14 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
15 # GNU General Public License for more details.
16 #
17 # You should have received a copy of the GNU General Public License
18 # along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
19
20 ## NB This file is mostly historic and predates the unit tests.
21 ##    Yet it still provides useful examples -- but the unitTests/
22 ##    for vastly larger coverage
23
24 suppressMessages(library(Rcpp))
25 suppressMessages(library(inline))
26
27 cat("===Doubles\n")
28 foo <- '
29         double d = Rcpp::as<double>(x);
30         std::cout << "Returning twice the value of " << d << " : ";
31         return(Rcpp::wrap( 2*d ) );
32         '
33 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
34 cat(funx(x=2.123), "\n")
35 cat(funx(x=2), "\n")
36 ##funx(x='2')  ## throws as expected
37 ##funx(x=2:3)  ## throws as expected
38
39
40 cat("\n===Int\n")
41 foo <- '
42         int i = Rcpp::as<int>(x);
43         std::cout << "Returning twice the value of " << i << " : ";
44         return(Rcpp::wrap( 2*i ) );
45         '
46 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
47 cat(funx(x=2), "\n")
48 cat(funx(x=2.2), "\n")
49 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
50 cat(funx(x=as.raw(2)), "\n")
51
52 cat("\n===String\n")
53 foo <- '
54         std::string s = Rcpp::as<std::string>(x);
55         std::cout << "Returning twice the value of " << s << " : ";
56         return(Rcpp::wrap( s+s ) );
57         '
58 funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
59 cat(funx(x="abc"), "\n")
60
61 cat("\n===Raw (bytes)\n")
62 foo <- '
63         Rbyte i = Rcpp::as<Rbyte>(x) ;
64         std::cout << "Returning twice the value of " << (int)i << " : ";
65         return(Rcpp::wrap( (Rbyte)(2*i) ) );
66         '
67 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
```

```
68 cat( funx(x=2), "\n")
69 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
70 cat( funx(x=2L), "\n")
71 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
72 cat( funx(x=as.raw(2)), "\n")
73
74 cat("\n=== logical \n")
75 foo <- '
76 bool b = Rcpp::as<bool>(x);
77 std::cout << "flip  " << ( b ? "TRUE" : "FALSE" ) << " : ";
78 return(Rcpp::wrap( !b ));
79 '
80 funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
81 cat( res <- funx(x=TRUE) , "\n")  ; stopifnot( !res )
82 cat( res <- funx(x=FALSE), "\n" ) ; stopifnot( res)
83 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
84 cat( res <- funx(x=2)  , "\n")   ; stopifnot( !res )
85 cat( res <- funx(x=0.0), "\n")   ; stopifnot( res)
86 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
87 cat( res <- funx(x=2L), "\n")    ; stopifnot( !res )
88 cat( res <- funx(x=0L), "\n")    ; stopifnot( res)
89 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
90 cat( res <- funx(x=as.raw(2)), "\n") ; stopifnot( !res )
91 cat( res <- funx(x=as.raw(0)), "\n") ; stopifnot( res)
92
93 ### vectors
94
95 cat("\n===Int Vector via wrap\n")
96 foo <- '
97         std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
98         std::cout << "Returning twice the value of vector : ";
99         for (size_t i=0; i<iv.size(); i++) {
100             iv[i] = 2*iv[i];
101         }
102         return(Rcpp::wrap(iv));
103         '
104 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
105 print(funx(x=2:5))
106 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
107 print(funx(x=2:5))
108 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
109 print(funx(x=as.raw(2:5)))
110
111 cat("\n===Int Vector\n")
112 foo <- '
113         std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
114         std::cout << "Returning twice the value of vector : ";
115         for (size_t i=0; i<iv.size(); i++) {
116             iv[i] = 2*iv[i];
117         }
118         return(Rcpp::wrap( iv ) );
119         '
120 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
121 print(funx(x=2:5+.1))
122 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
123 print(funx(x=2:5))
124 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
125 print(funx(x=as.raw(2:5)))
126
127
128 cat("\n===Double Vector\n")
129 foo <- '
130         std::vector<double> iv = Rcpp::as< std::vector<double> >(x) ;
131         std::cout << "Returning twice the value of vector : ";
132         for (size_t i=0; i<iv.size(); i++) {
133             iv[i] = 2*iv[i];
134         }
135         return(Rcpp::wrap( iv ));
136         '
137 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
138 print(funx(x=0.1+2:5))
139 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
140 print(funx(x=2:5))
141 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
142 print(funx(x=as.raw(2:5)))
143
144 cat("\n===Raw Vector\n")
145 foo <- '
146         std::vector<Rbyte> iv = Rcpp::as< std::vector<Rbyte> >(x) ;
147         std::cout << "Returning twice the value of vector : ";
148         for (size_t i=0; i<iv.size(); i++) {
```

```
149             iv[i] = 2*iv[i];
150         }
151         return(Rcpp::wrap( iv ));
152         '
153 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
154 print(funx(x=as.raw(0:9)))
155 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
156 print(funx(x=0:9))
157 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
158 print(funx(x=0:9+.1))
159
160 cat("\n=== vector<bool>\n")
161 foo <- '
162 std::vector<bool> bv = Rcpp::as< std::vector<bool> >(x) ;
163 std::cout << "Flip the value of vector : ";
164 for (size_t i=0; i<bv.size(); i++) {
165     bv[i].flip() ;
166 }
167 return(Rcpp::wrap( bv ));
168 '
169 funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
170 print(funx(x=c(TRUE,FALSE)))
171 funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
172 print(funx(x=as.raw(0:9)))
173 funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
174 print(funx(x=0:9))
175 funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
176 print(funx(x=as.numeric(0:9)))
177
178
179 cat("\n===String Vector\n")
180 foo <- '
181         std::vector<std::string> iv = Rcpp::as< std::vector<std::string> >(x);
182         std::cout << "Returning twice the value of vector : ";
183         for (size_t i=0; i<iv.size(); i++) {
184             iv[i] = iv[i] + iv[i];
185         }
186         return(Rcpp::wrap( iv ));
187         '
188 funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
189 print(funx(x=c("foo", "bar")))
190
191 ### using std::set
192 cat("\n=== set<int>\n")
193 foo <- '
194 std::set<int> iv ;
195 iv.insert( 0 ) ;
196 iv.insert( 1 ) ;
197 iv.insert( 0 ) ;
198 return Rcpp::wrap( iv );'
199 funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>" )
200 print(res <- funx())
201 stopifnot( identical( res, 0:1 ) )
202
203 cat("\n=== set<double>\n")
204 foo <- '
205 std::set<double> ds;
206 ds.insert( 0.0 );
207 ds.insert( 1.0 );
208 ds.insert( 0.0 );
209 return(Rcpp::wrap( ds )); '
210 funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>")
211 print( res <- funx() )
212 stopifnot( identical( res, as.numeric(0:1)))
213
214 cat("\n=== set<raw>\n")
215 foo <- '
216 std::set<Rbyte> bs ;
217 bs.insert( (Rbyte)0 ) ;
218 bs.insert( (Rbyte)1 ) ;
219 bs.insert( (Rbyte)0 ) ;
220 return(Rcpp::wrap( bs )); '
221 funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>")
222 print( res <- funx() )
223 stopifnot( identical( res, as.raw(0:1)))
224
225 cat("\n=== set<string> \n")
226 foo <- '
227 std::set<std::string> ss ;
228 ss.insert( "foo" ) ;
229 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::RObject(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::RObject(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::RObject(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::RObject(x).isNULL() ;
return Rcpp::wrap( is_null ) ;
’, Rcpp=TRUE, verbose=FALSE)
res <- funx( iris )
stopifnot( !res )
res <- funx( NULL )
stopifnot( res )


```

8.27 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.

```

```
27 require(inline)
28 funx_old <- cxxfunction(
29         signature(),
30         'throw std::range_error("boom"); return R_NilValue ; ',
31         plugin = "Rcpp" )
32
33 ## NOTE: Within this section, the new way to compile Rcpp code inline has been
34 ## written. Please use the code next as a template for your own project.
35
36 cppFunction('
37 SEXP funx(){
38     throw std::range_error("boom"); return R_NilValue ;
39 }')
40
41 tryCatch(  funx(), "C++Error" = function(e){
42         cat( sprintf( "C++ exception of class '%s' : %s\n", class(e)[1L], e$message  ) )
43 } )
44 # or using a direct handler
45 tryCatch(  funx(), "std::range_error" = function(e){
46         cat( sprintf( "C++ exception of class '%s' : %s\n", class(e)[1L], e$message  ) )
47 } )
48 # just to check things carry on
49 print( rnorm(10) )
50
```

8.28 SugarPerformance/sugarBenchmarks.R

Benchmarking the 'Rcpp sugar' extensions

```
1 #!/usr/bin/env r
2
3 suppressMessages(library(inline))
4 suppressMessages(library(Rcpp))
5
6 benchmark <- function(start = settings$start,
7                        hand.written = settings$hand.written,
8                        sugar = settings$sugar,
9                        expr = settings$expr,
10                        runs = settings$runs,
11                        data = settings$data,
12                        end = settings$end,
13                        inc = settings$inc,
14
15                        settings = list(
16                                  start = "", hand.written = "",
17                                  sugar = "", expr = NULL,
18                                  runs = 500,
19                                  data = NULL ,
20                                  end = "",
21                                  inc = ""
22                                  )
23                        ) {
24
25 expr <- force(expr)
26 inc  <- force( inc )
27
28 src <- sprintf( '
29     unsigned int runs = as<int>(runss);
30         Environment e(env) ;
31
32         %s
33
34     Timer timer;
35
36     // approach one
37     timer.Start();
38     for (unsigned int i=0; i<runs; i++) {
39           %s
40     }
41     timer.Stop();
42     double t1 = timer.ElapsedTime();
43
44     // approach two
```

```
45      timer.Reset(); timer.Start();
46      for (unsigned int i=0; i<runs; i++) {
47          %s
48      }
49      timer.Stop();
50      double t2 = timer.ElapsedTime();
51
52      Language call(expr) ;
53
54      timer.Reset(); timer.Start();
55      for (unsigned int i=0; i<runs; i++) {
56          NumericVector res2 = Rcpp_eval( call, e ) ;
57      }
58      timer.Stop();
59      double t3 = timer.ElapsedTime();
60
61      %s
62
63      return NumericVector::create(
64          _["hand written"] = t1,
65          _["sugar"] = t2,
66          _["R"]     = t3
67          ) ;
68  ',
69                 paste( start, collapse = "\n" ) ,
70                 paste( hand.written, collapse = "\n" ),
71                 paste( sugar, collapse = "\n" ),
72                 paste( end, collapse = "\n" )
73          )
74
75          e <- environment()
76          for( i in names(data) ){
77                  assign( i, data[[i]], envir = e )
78          }
79
80          settings <- getPlugin("Rcpp")
81          settings$env$PKG_CXXFLAGS <- paste("-I", getwd(), sep="")
82
83          fun <- cxxfunction(signature(runss="integer", expr = "language", env = "environment" ),
84                             src,
85                             includes= sprintf( '#include "Timer.h"\n%s', paste( inc, collapse = "\n" ) ),
86                             plugin="Rcpp",
87                             settings=settings)
88          results <- fun(runs, expr, environment() )
89          cat( "-" )
90          list( results = results, runs = runs, expr = deparse(expr) )
91  }
92
93  settings.ifelse <- list( start = '
94          NumericVector x = e["x"] ;
95          NumericVector y = e["y"] ;
96  ', hand.written = '
97          int n = x.size() ;
98          NumericVector res1( n ) ;
99          double x_ = 0.0 ;
100          double y_ = 0.0 ;
101          for( int i=0; i<n; i++){
102          x_ = x[i] ;
103          y_ = y[i] ;
104          if( R_IsNA(x_) || R_IsNA(y_) ){
105              res1[i] = NA_REAL;
106          } else if( x_ < y_ ){
107              res1[i] = x_ * x_ ;
108          } else {
109              res1[i] = -( y_ * y_)  ;
110          }
111      }
112
113  ', sugar = '
114      NumericVector res2 = ifelse( x < y, x*x, -(y*y) ) ;
115  ', expr = quote(ifelse(x<y, x*x, -(y*y) )),
116     data = list( x = runif(1e5),  y = runif(1e5) )
117  )
118
119  settings.ifelse.nona <- list( start = '
120          NumericVector x = e["x"] ;
121          NumericVector y = e["y"] ;
122  ', hand.written = '
123          int n = x.size() ;
124          NumericVector res1( n ) ;
125          double x_ = 0.0 ;
```

```
126         double y_ = 0.0 ;
127         for( int i=0; i<n; i++){
128         x_ = x[i] ;
129         y_ = y[i] ;
130         if( x_ < y_ ){
131             res1[i] = x_ * x_ ;
132         } else {
133             res1[i] = -( y_ * y_)  ;
134         }
135     }
136
137 ', sugar = '
138     NumericVector res2 = ifelse( x < y, noNA(x)*noNA(x), -(noNA(y)*noNA(y)) ) ;
139 ', expr = quote(ifelse(x<y, x*x, -(y*y) )),
140    data = list( x = runif(1e5),  y = runif(1e5) )
141 )
142
143 settings.sapply <- list( start =  '
144         NumericVector x = e["x"] ;
145         int n = x.size() ;
146
147 ', hand.written = '
148         NumericVector res1( n ) ;
149         std::transform( x.begin(), x.end(), res1.begin(), square ) ;
150
151 ', sugar = '
152         NumericVector res2 = sapply( x, square ) ;
153 ',
154         expr = quote(sapply(x,square)),
155         runs = 500,
156         data = list(
157                 x = rnorm(1e5) ,
158                 square = function(x) x*x
159         ),
160         inc = '
161         inline double square(double x){ return x*x ; }
162         '
163 )
164
165 settings.any <- list( start = '
166         NumericVector x = e["x"] ;
167         NumericVector y = e["y"] ;
168         int res ;
169         SEXP res2 ;
170
171 ', hand.written = '
172         int n = x.size() ;
173         bool seen_na = false ;
174         bool result = false ;
175         double x_ = 0.0 ;
176         double y_ = 0.0 ;
177         for( int i=0; i<n; i++){
178                 x_ = x[i] ;
179                 if( R_IsNA( x_ )  ){
180                         seen_na = true ;
181                 } else {
182                         y_ = y[i] ;
183                         if( R_IsNA( y_ ) ){
184                                 seen_na = true ;
185                         } else {
186                                 /* both non NA */
187                                 if( x_*y_ < 0.0 ){
188                                         result = true ;
189                                         break ;
190                                 }
191                         }
192                 }
193         }
194         res = result ? TRUE : ( seen_na ? NA_LOGICAL : FALSE ) ;
195 ', sugar = '
196         res2 = any( x*y < 0 ) ;
197 ',
198         expr = quote(any(x*y<0)),
199         runs = 5000,
200         data = list(
201                 x = seq( -1, 1, length = 1e05),
202                 y = rep( 1, 1e05)
203         )
204 )
205 raw.results <- list(
206         benchmark( settings = settings.any   , runs = 5000 ),
```

```
207         benchmark( settings = settings.ifelse, runs = 500 ),
208         benchmark( settings = settings.ifelse.nona, runs = 500 ),
209         benchmark( settings = settings.sapply, runs = 500 )
210 )
211 cat("\n")
212
213 results <- do.call( rbind, lapply( raw.results, "[[", "results" ) )
214 results <- data.frame(
215         runs = sapply( raw.results, "[[", "runs" ),
216         expr = sapply( raw.results, "[[", "expr" ),
217         as.data.frame( results, stringsAsFactors = FALSE )
218         )
219
220 results[[ "hand/sugar" ]] <- results[["hand.written" ]] / results[["sugar"]]
221 results[[ "R/sugar" ]]    <- results[["R" ]]            / results[["sugar"]]
222 # results <- results[ order( results[["expr"]], results[["runs"]] ), ]
223
224 options( width = 300 )
225 print( results )
226
```

8.29 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 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

#ifndef 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
```

8.30 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/fall03/

#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 << "Reseting" << 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;
}
```