Rcpp Version 1.0.9

Generated by Doxygen 1.9.1

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:

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

        - Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > 1824
    - Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive< RTYPE, NA, T > > 2561
        - Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T > 1779
    - Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE, NA, T > > 2561
        - Rcpp::sugar::Not_Vector< RTYPE, NA, T > 1700
    - Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > > 2561
        - Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T > 1557
    - Rcpp::VectorBase< RTYPE, true, Comparator_With_One_Value< LGLSXP, Operator, false, T > > 2561
        - Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T > 688
    - Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > > 2561
        - Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > 504
    - Rcpp::VectorBase< unary_minus_result_type< RTYPE >::value, NA, UnaryMinus_Vector< unary_↩ minus_result_type< RTYPE >::value, NA, T > > 2561
        - Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > 2465
    - Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > > 2561
        - Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > 2433
    - Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > > 2561
        - Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > 2320
    - Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > > 2561
        - Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > 1817
    - Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > 2561
        - Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > 1810
    - Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > > 2561
        - Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > 1730
    - Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS↩ _NA, RHS_T > > 2561
        - Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > 1720
    - Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > 2561
        - Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > 1589
    - Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > 2561
        - Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > 1585
    - Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, false, T > > 2561
        - Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T > 1565
    - Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, false, COND_T > > 2561
        - Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T > 1219
    - Rcpp::VectorBase< REALSXP, false, Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > 2561
        - Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > 1035
    - Rcpp::VectorBase< REALSXP, NA, Q3< RTYPE, NA, T > > 2561
        - Rcpp::stats::Q3< RTYPE, NA, T > 1910
    - Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > > 2561
        - Rcpp::stats::P2< RTYPE, NA, T > 1766
    - Rcpp::VectorBase< REALSXP, NA, D3< RTYPE, NA, T > > 2561
        - Rcpp::stats::D3< RTYPE, NA, T > 931
    - Rcpp::MatrixBase< RTYPE, na, MATRIX > 1494
    - Rcpp::VectorBase< RTYPE, na, VECTOR > 2561
        - Rcpp::Vector< VECSXP, StoragePolicy > 2494
- Rcpp::traits::expands_to_logical__impl< LGLSXP > 1090
- Rcpp::attributes::ExportsGenerator 1095

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 expm1 (double x)
- double log1p (double x)
- int imax2 (int x, int y)
- int imin2 (int x, int y)

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

5.1.1 Function Documentation

5.1.1.1 bessel_i()

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

Definition at line 209 of file Rmath.h.

5.1.1.2 bessel_i_ex()

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

Definition at line 213 of file Rmath.h.

5.1.1.3 bessel_j()

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

Definition at line 210 of file Rmath.h.

5.1.1.4 bessel_j_ex()

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

Definition at line 214 of file Rmath.h.

5.1.1.5 bessel_k()

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

Definition at line 211 of file Rmath.h.

5.1.1.6 bessel_k_ex()

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

Definition at line 215 of file Rmath.h.

5.1.1.7 bessel_y()

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

Definition at line 212 of file Rmath.h.

5.1.1.8 bessel_y_ex()

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

Definition at line 216 of file Rmath.h.

5.1.1.9 beta()

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

Definition at line 202 of file Rmath.h.

5.1.1.10 choose()

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

Definition at line 205 of file Rmath.h.

5.1.1.11 dbeta()

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

Definition at line 60 of file Rmath.h.

5.1.1.12 dbinom()

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

Definition at line 96 of file Rmath.h.

5.1.1.13 dcauchy()

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

Definition at line 105 of file Rmath.h.

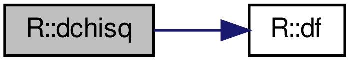
5.1.1.14 dchisq()

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

Definition at line 72 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.15 dexp()

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

Definition at line 111 of file Rmath.h.

5.1.1.16 df()

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

Definition at line 84 of file Rmath.h.

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

5.1.1.17 dgamma()

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

Definition at line 48 of file Rmath.h.

5.1.1.18 dgeom()

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

Definition at line 117 of file Rmath.h.

5.1.1.19 dhyper()

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

Definition at line 123 of file Rmath.h.

5.1.1.20 digamma()

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

Definition at line 197 of file Rmath.h.

5.1.1.21 dlnorm()

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

Definition at line 66 of file Rmath.h.

5.1.1.22 dlogis()

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

Definition at line 154 of file Rmath.h.

5.1.1.23 dnbeta()

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

Definition at line 160 of file Rmath.h.

5.1.1.24 dnbinom()

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

Definition at line 129 of file Rmath.h.

5.1.1.25 dnbinom_mu()

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

Definition at line 135 of file Rmath.h.

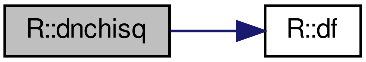
5.1.1.26 dnchisq()

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

Definition at line 78 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.27 dnf()

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

Definition at line 166 of file Rmath.h.

5.1.1.28 dnorm()

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

Definition at line 35 of file Rmath.h.

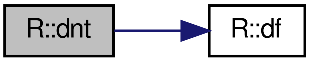
5.1.1.29 dnt()

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

Definition at line 171 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.30 dpois()

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

Definition at line 142 of file Rmath.h.

5.1.1.31 dpsifn()

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

Definition at line 195 of file Rmath.h.

5.1.1.32 dsignrank()

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

Definition at line 186 of file Rmath.h.

5.1.1.33 dt()

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

Definition at line 90 of file Rmath.h.

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

5.1.1.34 dunif()

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

Definition at line 42 of file Rmath.h.

5.1.1.35 dweibull()

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

Definition at line 148 of file Rmath.h.

5.1.1.36 dwilcox()

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

Definition at line 180 of file Rmath.h.

5.1.1.37 exp_rand()

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

Definition at line 32 of file Rmath.h.

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

5.1.1.38 expm1()

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

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

5.1.1.39 fmax2()

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

Definition at line 232 of file Rmath.h.

5.1.1.40 fmin2()

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

Definition at line 233 of file Rmath.h.

5.1.1.41 fprec()

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

Definition at line 235 of file Rmath.h.

5.1.1.42 fround()

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

Definition at line 236 of file Rmath.h.

5.1.1.43 fsign()

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

Definition at line 237 of file Rmath.h.

5.1.1.44 ftrunc()

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

Definition at line 238 of file Rmath.h.

5.1.1.45 gammafn()

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

Definition at line 192 of file Rmath.h.

5.1.1.46 hypot()

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

Definition at line 220 of file Rmath.h.

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

5.1.1.47 imax2()

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

Definition at line 230 of file Rmath.h.

5.1.1.48 imin2()

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

Definition at line 231 of file Rmath.h.

5.1.1.49 lbeta()

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

Definition at line 203 of file Rmath.h.

5.1.1.50 lchoose()

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

Definition at line 206 of file Rmath.h.

5.1.1.51 lgamma1p()

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

Definition at line 55 of file Rmath.h.

5.1.1.52 lgammafn()

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

Definition at line 193 of file Rmath.h.

5.1.1.53 lgammafn_sign()

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

Definition at line 194 of file Rmath.h.

5.1.1.54 log1p()

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

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

5.1.1.55 log1pexp()

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

Definition at line 54 of file Rmath.h.

5.1.1.56 log1pmx()

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

Definition at line 53 of file Rmath.h.

5.1.1.57 logspace_add()

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

Definition at line 56 of file Rmath.h.

5.1.1.58 logspace_sub()

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

Definition at line 57 of file Rmath.h.

5.1.1.59 norm_rand()

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

Definition at line 30 of file Rmath.h.

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

5.1.1.60 pbeta()

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

Definition at line 61 of file Rmath.h.

5.1.1.61 pbinom()

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

Definition at line 97 of file Rmath.h.

5.1.1.62 pcauchy()

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

Definition at line 106 of file Rmath.h.

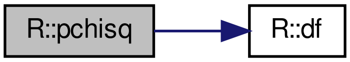
5.1.1.63 pchisq()

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

Definition at line 73 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.64 pentagamma()

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

Definition at line 200 of file Rmath.h.

5.1.1.65 pexp()

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

Definition at line 112 of file Rmath.h.

5.1.1.66 pf()

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

Definition at line 85 of file Rmath.h.

5.1.1.67 pgamma()

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

Definition at line 49 of file Rmath.h.

5.1.1.68 pgeom()

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

Definition at line 118 of file Rmath.h.

5.1.1.69 phyper()

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

Definition at line 124 of file Rmath.h.

5.1.1.70 plnorm()

```
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 plogis()

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

Definition at line 155 of file Rmath.h.

5.1.1.72 pnbeta()

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

Definition at line 161 of file Rmath.h.

5.1.1.73 pnbinom()

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

Definition at line 130 of file Rmath.h.

5.1.1.74 pnbinom_mu()

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

Definition at line 136 of file Rmath.h.

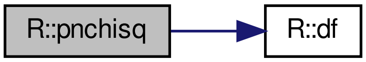
5.1.1.75 pnchisq()

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

Definition at line 79 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.76 pnf()

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

Definition at line 167 of file Rmath.h.

5.1.1.77 pnorm()

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

Definition at line 36 of file Rmath.h.

5.1.1.78 pnorm_both()

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

Definition at line 39 of file Rmath.h.

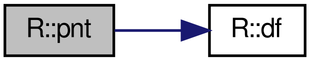
5.1.1.79 pnt()

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

Definition at line 172 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.80 ppois()

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

Definition at line 143 of file Rmath.h.

5.1.1.81 psigamma()

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

Definition at line 196 of file Rmath.h.

5.1.1.82 psignrank()

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

Definition at line 187 of file Rmath.h.

5.1.1.83 pt()

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

Definition at line 91 of file Rmath.h.

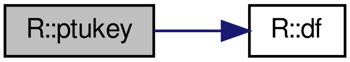
5.1.1.84 ptukey()

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

Definition at line 176 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.85 punif()

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

Definition at line 43 of file Rmath.h.

5.1.1.86 pweibull()

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

Definition at line 149 of file Rmath.h.

5.1.1.87 pwilcox()

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

Definition at line 181 of file Rmath.h.

5.1.1.88 qbeta()

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

Definition at line 62 of file Rmath.h.

5.1.1.89 qbinom()

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

Definition at line 98 of file Rmath.h.

5.1.1.90 qcauchy()

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

Definition at line 107 of file Rmath.h.

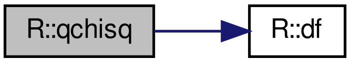
5.1.1.91 qchisq()

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

Definition at line 74 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.92 qexp()

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

Definition at line 113 of file Rmath.h.

5.1.1.93 qf()

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

Definition at line 86 of file Rmath.h.

5.1.1.94 qgamma()

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

Definition at line 50 of file Rmath.h.

5.1.1.95 qgeom()

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

Definition at line 119 of file Rmath.h.

5.1.1.96 qhyper()

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

Definition at line 125 of file Rmath.h.

5.1.1.97 qlnorm()

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

Definition at line 68 of file Rmath.h.

5.1.1.98 qlogis()

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

Definition at line 156 of file Rmath.h.

5.1.1.99 qnbeta()

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

Definition at line 162 of file Rmath.h.

5.1.1.100 qnbinom()

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

Definition at line 131 of file Rmath.h.

5.1.1.101 qnbinom_mu()

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

Definition at line 137 of file Rmath.h.

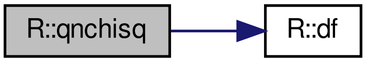
5.1.1.102 qnchisq()

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

Definition at line 80 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.103 qnf()

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

Definition at line 168 of file Rmath.h.

5.1.1.104 qnorm()

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

Definition at line 37 of file Rmath.h.

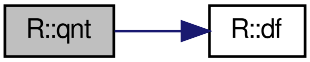
5.1.1.105 qnt()

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

Definition at line 173 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.106 qpois()

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

Definition at line 144 of file Rmath.h.

5.1.1.107 qsignrank()

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

Definition at line 188 of file Rmath.h.

5.1.1.108 qt()

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

Definition at line 92 of file Rmath.h.

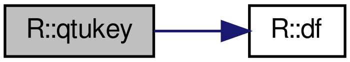
5.1.1.109 qtukey()

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

Definition at line 177 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.110 qunif()

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

Definition at line 44 of file Rmath.h.

5.1.1.111 qweibull()

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

Definition at line 150 of file Rmath.h.

5.1.1.112 qwilcox()

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

Definition at line 182 of file Rmath.h.

5.1.1.113 rbeta()

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

Definition at line 63 of file Rmath.h.

5.1.1.114 rbinom()

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

Definition at line 99 of file Rmath.h.

5.1.1.115 rcauchy()

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

Definition at line 108 of file Rmath.h.

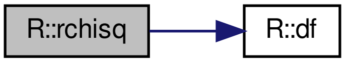
5.1.1.116 rchisq()

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

Definition at line 75 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.117 rexp()

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

Definition at line 114 of file Rmath.h.

5.1.1.118 rf()

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

Definition at line 87 of file Rmath.h.

5.1.1.119 rgamma()

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

Definition at line 51 of file Rmath.h.

5.1.1.120 rgeom()

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

Definition at line 120 of file Rmath.h.

5.1.1.121 rhyper()

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

Definition at line 126 of file Rmath.h.

5.1.1.122 rlnorm()

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

Definition at line 69 of file Rmath.h.

5.1.1.123 rlogis()

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

Definition at line 157 of file Rmath.h.

5.1.1.124 rmultinom()

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

Definition at line 102 of file Rmath.h.

5.1.1.125 rnbeta()

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

Definition at line 163 of file Rmath.h.

5.1.1.126 rnbinom()

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

Definition at line 132 of file Rmath.h.

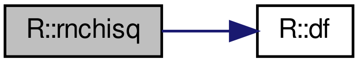
5.1.1.127 rnchisq()

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

Definition at line 81 of file Rmath.h.

References df().

Here is the call graph for this function:



5.1.1.128 rnorm()

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

Definition at line 38 of file Rmath.h.

5.1.1.129 rpois()

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

Definition at line 145 of file Rmath.h.

5.1.1.130 rsignrank()

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

Definition at line 189 of file Rmath.h.

5.1.1.131 rt()

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

Definition at line 93 of file Rmath.h.

5.1.1.132 runif()

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

Definition at line 45 of file Rmath.h.

5.1.1.133 rweibull()

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

Definition at line 151 of file Rmath.h.

5.1.1.134 rwilcox()

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

Definition at line 183 of file Rmath.h.

5.1.1.135 sign()

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

Definition at line 234 of file Rmath.h.

5.1.1.136 tetragamma()

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

Definition at line 199 of file Rmath.h.

5.1.1.137 trigamma()

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

Definition at line 198 of file Rmath.h.

5.1.1.138 unif_rand()

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

Definition at line 31 of file Rmath.h.

Referenced by Rcpp::sugar::EmpiricalSample(), Rcpp::stats::CauchyGenerator::operator()(), Rcpp::stats::Cauchy↩ Generator_1::operator()(), Rcpp::stats::CauchyGenerator_0::operator()(), Rcpp::stats::LogisGenerator::operator()(), Rcpp::stats::LogisGenerator_1::operator()(), Rcpp::stats::LogisGenerator_0::operator()(), Rcpp::stats::UnifGenerator↩ ::operator()(), Rcpp::stats::UnifGenerator__0__1::operator()(), Rcpp::stats::WeibullGenerator::operator()(), Rcpp↩ ::stats::WeibullGenerator__scale1::operator()(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), and Rcpp::sugar::WalkerSample().

5.2 Rcpp Namespace Reference

Rcpp API.

Namespaces

- algorithm
- attributes
- internal

  internal implementation details
- InternalFunctionWithStdFunction
- stats
- sugar
- traits

  traits used to dispatch wrap

Classes

- class Timer
- class DataFrame_Impl
- class Date
- class Datetime
- class newDatetimeVector
- class newDateVector
- class oldDatetimeVector
- class oldDateVector
- class Dimension
- class DottedPairImpl
- class exception
- class no_such_env
- class file_io_error
- class file_not_found

- class file_exists
- struct LongjumpException
- class Fast
- class InputParameter
- class ReferenceInputParameter
- class ConstInputParameter
- class ConstReferenceInputParameter
- class Rstreambuf
- class Rostream
- class fixed_call
- class unary_call
- class binary_call
- class class_Base
- class CppFunctionBase
- class CppFunction
- struct void_type
- class Module
- class result
- class object
- class CppMethod
- class CppInheritedMethod
- class SignedConstructor
- class SignedFactory
- class SignedMethod
- class S4_CppConstructor
- class S4_CppOverloadedMethods
- class CppProperty
- class CppInheritedProperty
- class CppFinalizer
- class FunctionFinalizer
- class S4_field
- class enum_
- class CppClass
- class CppObject
- class Na_Proxy
- class Argument
- class Nullable
- class Armor
- class Shelter
- class Shield
- class AttributeProxyPolicy
- class BindingPolicy
- class DottedPairProxyPolicy
- class FieldProxyPolicy
- struct GenericProxy
- class NamesProxyPolicy
- class ProtectedProxyPolicy
- class RObjectMethods
- class SlotProxyPolicy
- class TagProxyPolicy
- class RNGScope

- class SuspendRNGSynchronizationScope
- class Generator
- class NoProtectStorage
- class PreserveStorage
- class String
- class StringTransformer
- struct can_have_na
- class Range
- class Vector
- class ChildVector
- class ListOf
- class Matrix
- class MatrixBase
- class MatrixColumn
- class ConstMatrixColumn
- class MatrixRow
- class ConstMatrixRow
- class no_init_vector
- class no_init_matrix
- class SubMatrix
- class SubsetProxy
- class VectorBase
- class SingleLogicalResult
- class XPtr
- struct tzhead
- struct ttinfo
- struct lsinfo
- struct state
- struct rule

Typedefs

- typedef uint64_t nanotime_t
- typedef DataFrame_Impl< PreserveStorage > DataFrame
- typedef oldDateVector DateVector
- typedef oldDatetimeVector DatetimeVector
- typedef DottedPair_Impl< PreserveStorage > DottedPair
- typedef Environment_Impl< PreserveStorage > Environment
- typedef Formula_Impl< PreserveStorage > Formula
- typedef Function_Impl< PreserveStorage > Function
- typedef InternalFunction_Impl< PreserveStorage > InternalFunction
- typedef DottedPairProxyPolicy< Language_Impl >::const_DottedPairProxy const_Proxy
- typedef Language_Impl< PreserveStorage > Language
- typedef bool(∗ ValidConstructor) (SEXP ∗, int)
- typedef bool(∗ ValidMethod) (SEXP ∗, int)
- typedef Pairlist_Impl< PreserveStorage > Pairlist
- typedef Promise_Impl< PreserveStorage > Promise
- typedef Reference_Impl< PreserveStorage > Reference
- typedef RObject_Impl< PreserveStorage > RObject
- typedef S4_Impl< PreserveStorage > S4

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

Functions

- template<class InputIterator , class T >
  bool __any (InputIterator first, InputIterator last, const T &value, std::input_iterator_tag)
- template<class RandomAccessIterator , class T >
  bool __any (RandomAccessIterator __first, RandomAccessIterator __last, const T &__val, std::random_access↩_iterator_tag)
- template<class InputIterator , class T >
  bool any (InputIterator first, InputIterator last, const T &value)
- template<class InputIterator , class Predicate >
  bool __any_if (InputIterator first, InputIterator last, Predicate pred, std::input_iterator_tag)
- template<class RandomAccessIterator , class Predicate >
  bool __any_if (RandomAccessIterator __first, RandomAccessIterator __last, Predicate __pred, std::random_↩access_iterator_tag)
- template<class InputIterator , class Predicate >
  bool any_if (InputIterator first, InputIterator last, Predicate pred)
- template<> SEXP wrap_extra_steps< Rcpp::Date > (SEXP)
- template<> SEXP wrap_extra_steps< Rcpp::Datetime > (SEXP)
- template<typename T >
  SEXP wrap_extra_steps (SEXP x)
- template<> SEXP wrap (const Date &date)
- template<> SEXP wrap< Datetime > (const Datetime &date)
- Environment new_env (int size=29)
- Environment new_env (SEXP parent, int size=29)
- void message (SEXP s)

- SEXP Rcpp_fast_eval (SEXP expr, SEXP env)
- SEXP Rcpp_eval (SEXP expr, SEXP env)
- template<typename T >
  T as (SEXP x)
- template<> char as< char > (SEXP x)
- template<typename T >
  traits::remove_const_and_reference< T >::type bare_as (SEXP x)
- template<> SEXP as (SEXP x)
- template<typename T >
  T clone (const T &object)
- template<> SEXP wrap< Rcpp::Date > (const Rcpp::Date &date)
- Date operator+ (const Date &date, int offset)
- double operator- (const Date &d1, const Date &d2)
- bool operator< (const Date &d1, const Date &d2)
- bool operator> (const Date &d1, const Date &d2)
- bool operator== (const Date &d1, const Date &d2)
- bool operator>= (const Date &d1, const Date &d2)
- bool operator<= (const Date &d1, const Date &d2)
- bool operator!= (const Date &d1, const Date &d2)
- std::ostream & operator<< (std::ostream &os, const Date d)
- template<> SEXP wrap< Rcpp::Datetime > (const Rcpp::Datetime &dt)
- Datetime operator+ (const Datetime &datetime, double offset)
- Datetime operator+ (const Datetime &datetime, int offset)
- double operator- (const Datetime &d1, const Datetime &d2)
- bool operator< (const Datetime &d1, const Datetime &d2)
- bool operator> (const Datetime &d1, const Datetime &d2)
- bool operator== (const Datetime &d1, const Datetime &d2)
- bool operator>= (const Datetime &d1, const Datetime &d2)
- bool operator<= (const Datetime &d1, const Datetime &d2)
- bool operator!= (const Datetime &d1, const Datetime &d2)
- std::ostream & operator<< (std::ostream &os, const Datetime d)
- std::ostream & operator<< (std::ostream &os, const newDatetimeVector d)
- std::ostream & operator<< (std::ostream &os, const newDateVector d)
- RCPP_API_CLASS (DottedPair_Impl)
- RCPP_API_CLASS (Environment_Impl)
- Environment_Impl (SEXP x)
- Environment_Impl (const std::string &name)
- Environment_Impl (int pos)
- SEXP ls (bool all) const
- SEXP get (const std::string &name) const
- SEXP get (Symbol name) const
- SEXP find (const std::string &name) const
- SEXP find (Symbol name) const
- bool exists (const std::string &name) const
- bool assign (const std::string &name, SEXP x) const
- bool assign (const std::string &name, const Shield< SEXP > &x) const
- template<typename WRAPPABLE >
  bool assign (const std::string &name, const WRAPPABLE &x) const
- bool isLocked () const
- bool remove (const std::string &name)
- void lock (bool bindings=false)

- void lockBinding (const std::string &name)
- void unlockBinding (const std::string &name)
- bool 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) const
- void update (SEXP)
- template<typename... Args>
  void warning (const char ∗fmt, Args &&... args)
- template<typename... Args>
  void NORET stop (const char ∗fmt, Args &&... args)
- template<typename T1 >
  void warning (const char ∗fmt, const T1 &arg1)
- template<typename T1 , typename T2 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1 , typename T2 , typename T3 , typename T4 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >
  void warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- template<typename T1 >
  void NORET stop (const char ∗fmt, const T1 &arg1)
- template<typename T1 , typename T2 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)

- template<typename T1 , typename T2 , typename T3 , typename T4 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >
  void NORET stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- static std::string toString (const int i)
- void warning (const std::string &message)
- void NORET stop (const std::string &message)
- RCPP_SIMPLE_EXCEPTION_CLASS (not_a_matrix, "Not a matrix.") RCPP_SIMPLE_EXCEPTION_↩
  CLASS(no_such_field
- No such field RCPP_EXCEPTION_CLASS (reference_creation_error, "Error creating object of reference class") RCPP_ADVANCED_EXCEPTION_CLASS(not_compatible
- RCPP_API_CLASS (Formula_Impl)
- Formula_Impl (SEXP x)
- Formula_Impl (const std::string &code)
- RCPP_API_CLASS (Function_Impl)
- SEXP pairlist ()
- SEXP grow (SEXP head, SEXP tail)
- template<typename T >
  SEXP grow (const T &head, SEXP tail)
- SEXP grow (const char ∗head, SEXP tail)
- RCPP_API_CLASS (InternalFunction_Impl)
- void checkUserInterrupt ()
- template<typename T >
  bool is (SEXP x)
- SEXP Rcpp_list2 (SEXP x0, SEXP x1)
- SEXP Rcpp_list3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp_list4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP Rcpp_list5 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4)
- SEXP Rcpp_list6 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5)
- SEXP Rcpp_list7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP Rcpp_lang7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP Rcpp_list8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP Rcpp_lang8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP Rcpp_list9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)

- 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)
- void Rcpp_unprotect (int i)
- template<int TARGET>
  SEXP r_cast (SEXP x)
- RCPP_API_CLASS (Reference_Impl)
- RCPP_API_CLASS (RObject_Impl)
- attribute_hidden const char ∗ type2name (SEXP x)
- attribute_hidden double mktime00 (struct tm &tm)
- attribute_hidden struct tm ∗ gmtime_ (const time_t ∗const x)
- attribute_hidden void Rcpp_precious_init ()
- attribute_hidden void Rcpp_precious_teardown ()
- attribute_hidden SEXP Rcpp_precious_preserve (SEXP object)
- attribute_hidden void Rcpp_precious_remove (SEXP token)
- attribute_hidden Rostream< true > & Rcpp_cout_get ()
- attribute_hidden Rostream< false > & Rcpp_cerr_get ()

- RCPP_API_CLASS (S4_Impl)
- template<int MAX_SIZE>
  std::string sprintf (const char ∗format,...)
- template<bool NA, typename T >
  stats::D1< REALSXP, NA, T > dexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool log=false)
- template<bool NA, typename T >
  stats::P1< REALSXP, NA, T > pexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)
- template<bool NA, typename T >
  stats::Q1< REALSXP, NA, T > qexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)
- NumericVector rnorm (int n, double mean, double sd)
- NumericVector rnorm (int n, double mean)
- NumericVector rnorm (int n)
- NumericVector rbeta (int n, double a, double b)
- NumericVector rbinom (int n, double nin, double pp)
- NumericVector rcauchy (int n, double location, double scale)
- NumericVector rcauchy (int n, double location)
- NumericVector rcauchy (int n)
- NumericVector rchisq (int n, double df)
- NumericVector rexp (int n, double rate)
- NumericVector rexp (int n)
- NumericVector rf (int n, double n1, double n2)
- NumericVector rgamma (int n, double a, double scale)
- NumericVector rgamma (int n, double a)
- NumericVector rgeom (int n, double p)
- NumericVector rhyper (int n, double nn1, double nn2, double kk)
- NumericVector rlnorm (int n, double meanlog, double sdlog)
- NumericVector rlnorm (int n, double meanlog)
- NumericVector rlnorm (int n)
- NumericVector rlogis (int n, double location, double scale)
- NumericVector rlogis (int n, double location)
- NumericVector rlogis (int n)
- NumericVector 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< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, NA, T > &vec, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<bool NA, typename T >
  sugar::Cummax< INTSXP, NA, T > cummax (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cummax< REALSXP, NA, T > cummax (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cummin< INTSXP, NA, T > cummin (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cummin< REALSXP, NA, T > cummin (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumprod< INTSXP, NA, T > cumprod (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumprod< REALSXP, NA, T > cumprod (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumprod< CPLXSXP, NA, T > cumprod (const VectorBase< CPLXSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumsum< INTSXP, NA, T > cumsum (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumsum< REALSXP, NA, T > cumsum (const VectorBase< REALSXP, NA, T > &t)
- template<bool LHS_NA, typename LHS_T >
  sugar::Diff< INTSXP, LHS_NA, LHS_T > diff (const VectorBase< INTSXP, LHS_NA, LHS_T > &lhs)
- template<bool LHS_NA, typename LHS_T >
  sugar::Diff< REALSXP, LHS_NA, LHS_T > diff (const VectorBase< REALSXP, LHS_NA, LHS_T > &lhs)
- template<int RTYPE, bool NA, typename T >
  LogicalVector duplicated (const VectorBase< RTYPE, NA, T > &x)
- template<int RTYPE, bool NA, typename T >
  sugar::Head< RTYPE, NA, T > head (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)

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

- template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
  IntegerVector match (const VectorBase< RTYPE, NA, T > &x, const VectorBase< RTYPE, RHS_NA, RHS_T > &table_)
- template<int RTYPE, bool NA, typename T >
  sugar::Max< RTYPE, NA, T > max (const VectorBase< RTYPE, NA, T > &x)
- template<bool NA, typename T >
  sugar::Mean< REALSXP, NA, T > mean (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Mean< INTSXP, NA, T > mean (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Mean< CPLXSXP, NA, T > mean (const VectorBase< CPLXSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Mean< LGLSXP, NA, T > mean (const VectorBase< LGLSXP, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  sugar::median_detail::result< RTYPE >::type median (const Rcpp::VectorBase< RTYPE, NA, T > &x, bool na↩_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::Min< RTYPE, NA, T > min (const VectorBase< RTYPE, NA, T > &x)
- template<int RTYPE, bool NA, typename T >
  Vector< RTYPE > na_omit (const VectorBase< RTYPE, NA, T > &t)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > pmax (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > pmax (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool RHS_NA, typename RHS_T >
  sugar::Pmax_Vector_Primitive< RTYPE, RHS_NA, RHS_T > pmax (typename Rcpp::traits::storage_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > pmin (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > pmin (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool RHS_NA, typename RHS_T >
  sugar::Pmin_Vector_Primitive< RTYPE, RHS_NA, RHS_T > pmin (typename Rcpp::traits::storage_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
  sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > pow (const VectorBase< RTYPE, NA, T > &t, EXPONENT↩_TYPE exponent)
- template<int RTYPE, bool NA, typename T >
  sugar::Range< RTYPE, NA, T > range (const VectorBase< RTYPE, NA, T > &x)
- template<int RTYPE, bool NA, typename T >
  sugar::Rep< RTYPE, NA, T > rep (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)
- sugar::Rep_Single< double > rep (const double &x, R_xlen_t n)
- sugar::Rep_Single< int > rep (const int &x, R_xlen_t n)
- sugar::Rep_Single< Rbyte > rep (const Rbyte &x, R_xlen_t n)
- sugar::Rep_Single< Rcomplex > rep (const Rcomplex &x, R_xlen_t n)
- sugar::Rep_Single< bool > rep (const bool &x, R_xlen_t n)
- template<int RTYPE, bool NA, typename T >
  sugar::Rep_each< RTYPE, NA, T > rep_each (const VectorBase< RTYPE, NA, T > &t, R_xlen_t times)

- template<int RTYPE, bool NA, typename T >
  sugar::Rep_len< RTYPE, NA, T > rep_len (const VectorBase< RTYPE, NA, T > &t, R_xlen_t len)
- template<int RTYPE, bool NA, typename T >
  sugar::Rev< RTYPE, NA, T > rev (const VectorBase< RTYPE, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::RowSumsReturn< RTYPE >::type rowSums (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::ColSumsReturn< RTYPE >::type colSums (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::RowMeansReturn< RTYPE >::type rowMeans (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::ColMeansReturn< RTYPE >::type colMeans (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- Vector< INTSXP > sample (int n, int size, bool replace=false, sugar::probs_t probs=R_NilValue, bool one_based=true)
- template<int RTYPE>
  Vector< RTYPE > sample (const Vector< RTYPE > &x, int size, bool replace=false, sugar::probs_t probs=R_NilValue)
- template<int RTYPE, bool NA, typename T , typename Function >
  sugar::Sapply< RTYPE, NA, T, Function, traits::same_type< typename ::Rcpp::sugar::sapply_application_result_of< Function, T >::type, typename Rcpp::traits::storage_type< traits::r_sexptype_traits< typename ::Rcpp::sugar::sapply_application_r
  Function, T >::type >::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)
- Vector< STRSXP > trimws (const Vector< STRSXP > &x, const char ∗which="both")

- Matrix< STRSXP > trimws (const Matrix< STRSXP > &x, const char ∗which="both")
- String trimws (const String &str, const char ∗which="both")
- template<bool NA, typename T >
  sugar::Sum< INTSXP, NA, T > sum (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Sum< REALSXP, NA, T > sum (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Sum< LGLSXP, NA, T > sum (const VectorBase< LGLSXP, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  IntegerVector table (const VectorBase< RTYPE, NA, T > &x)
- template<int RTYPE, bool NA, typename T >
  sugar::Tail< RTYPE, NA, T > tail (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)
- template<int RTYPE, bool NA, typename T >
  Vector< RTYPE > unique (const VectorBase< RTYPE, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  Vector< RTYPE > sort_unique (const VectorBase< RTYPE, NA, T > &t, bool decreasing=false)
- template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
  LogicalVector in (const VectorBase< RTYPE, NA, T > &x, const VectorBase< RTYPE, RHS_NA, RHS_T > &table)
- template<bool NA, typename T >
  sugar::Var< REALSXP, NA, T > var (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Var< INTSXP, NA, T > var (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Var< LGLSXP, NA, T > var (const VectorBase< LGLSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Var< CPLXSXP, NA, T > var (const VectorBase< CPLXSXP, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  R_xlen_t which_max (const VectorBase< RTYPE, NA, T > &t)
- template<int RTYPE, bool NA, typename T >
  R_xlen_t which_min (const VectorBase< RTYPE, NA, T > &t)
- template<bool NA, typename T >
  bool is_true (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool is_false (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool is_na (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > as_vector (const MatrixBase< RTYPE, NA, T > &t)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Col< RTYPE, LHS_NA, LHS_T > col (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > &lhs)
- template<typename T >
  sugar::diag_result_type_trait< T >::type diag (const T &t)
- template<int RTYPE, bool NA, typename T >
  sugar::LowerTri< RTYPE, NA, T > lower_tri (const Rcpp::MatrixBase< RTYPE, NA, T > &lhs, bool diag=false)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function >
  sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > outer (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs, Function fun)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Row< RTYPE, LHS_NA, LHS_T > row (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > &lhs)
- template<int RTYPE, bool NA, typename T >
  sugar::UpperTri< RTYPE, NA, T > upper_tri (const Rcpp::MatrixBase< RTYPE, NA, T > &lhs, bool diag=false)

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

- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > >::type operator∗ (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > >::type operator∗ (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > operator∗ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- RCPP_API_CLASS (Symbol_Impl)
- SEXP unwindProtect (SEXP(∗callback)(void ∗data), void ∗data)
- template<typename T , typename Function >
  List lapply (const ListOf< T > &t, Function fun)
- template<typename T , typename Function >
  T sapply (const ListOf< T > &t, Function fun)
- internal::DimNameProxy rownames (SEXP x)
- internal::DimNameProxy colnames (SEXP x)
- template<template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Matrix< REALSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Matrix< INTSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Matrix< STRSXP, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Matrix< RTYPE, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>
  Matrix< RTYPE, StoragePolicy > 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 (R_xlen_t size)
- no_init_matrix no_init (int nr, int nc)
- template<int RTYPE, template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Vector< RTYPE, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &s, const Vector< STRSXP, StoragePolicy > &rhs)
- RCPP_API_CLASS (WeakReference_Impl)
- template<typename T >
  void standard_delete_finalizer (T ∗obj)
- template<typename T , void Finalizer>
  void finalizer_wrapper (SEXP p)
- SEXP Rcpp_PreserveObject (SEXP x)
- void Rcpp_ReleaseObject (SEXP x)
- SEXP Rcpp_ReplaceObject (SEXP x, SEXP y)
- SEXP Rcpp_PreciousPreserve (SEXP object)
- void Rcpp_PreciousRelease (SEXP token)
- static int tzparse (const char ∗name, struct state ∗sp, int lastditch)

- static int typesequiv (const struct state ∗sp, int a, int b)
- static const char ∗ getsecs (const char ∗strp, int_fast32_t ∗secsp)
- static const char ∗ getnum (const char ∗strp, int ∗const nump, const int min, const int max)
- static const char ∗ getrule (const char ∗strp, struct rule ∗const rulep)
- static int_fast32_t transtime (int year, const struct rule ∗rulep, int_fast32_t offset)
- static struct tm ∗ timesub (const time_t ∗timep, int_fast32_t offset, const struct state ∗sp, struct tm ∗tmp)
- static int leaps_thru_end_of (const int y)
- static int increment_overflow (int ∗const ip, int j)
- static int increment_overflow_time (time_t ∗tp, int_fast32_t j)
- static int_fast32_t detzcode (const char ∗const codep)
- static int_fast64_t detzcode64 (const char ∗const codep)
- static int differ_by_repeat (const time_t t1, const time_t t0)
- static const char ∗ getzname (const char ∗strp)
- static const char ∗ getqzname (const char ∗strp, const int delim)
- static const char ∗ getoffset (const char ∗strp, int_fast32_t ∗const offsetp)
- static int tzload (const char ∗name, struct state ∗const sp, const int doextend)
- static void gmtload (struct state ∗const sp)
- static struct tm ∗ gmtsub (const time_t ∗const timep, const int_fast32_t offset, struct tm ∗const tmp)

Variables

- public DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >
- public public DottedPairImpl< DottedPair_Impl< StoragePolicy > >
- public BindingPolicy< Environment_Impl< StoragePolicy > >
- public DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >
- public public DottedPairImpl< Formula_Impl< StoragePolicy > >
- static Rostream< true > Rcout = Rcpp_cout_get()
- static Rostream< false > Rcerr = Rcpp_cerr_get()
- public DottedPairProxyPolicy< Language_Impl< StoragePolicy > >
- public public DottedPairImpl< Language_Impl< StoragePolicy > >
- static Na_Proxy NA
- static internal::NamedPlaceHolder _
- public DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >
- public public DottedPairImpl< Pairlist_Impl< StoragePolicy > >
- public FieldProxyPolicy< Reference_Impl< StoragePolicy > >
- public DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > >
- static SEXP Rcpp_precious = R_NilValue
- static time_t const time_t_min = MINVAL(time_t, TYPE_BIT(time_t))
- static time_t const time_t_max = MAXVAL(time_t, TYPE_BIT(time_t))
- static const char gmt [ ] = "GMT"
- static const int mon_lengths [2][MONSPERYEAR]
- static const int year_lengths [2]
- static int gmt_is_set
- static struct state gmtmem
- static struct tm tm
- static Module ∗ current_scope

5.2.1 Detailed Description

Rcpp API.

5.2.2 Typedef Documentation

5.2.2.1 CharacterMatrix

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

Definition at line 46 of file instantiation.h.

5.2.2.2 CharacterVector

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

Definition at line 34 of file instantiation.h.

5.2.2.3 ComplexMatrix

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

Definition at line 40 of file instantiation.h.

5.2.2.4 ComplexVector

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

Definition at line 27 of file instantiation.h.

5.2.2.5 const_Proxy

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

Definition at line 38 of file Language.h.

5.2.2.6 DataFrame

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

Definition at line 190 of file DataFrame.h.

5.2.2.7 DatetimeVector

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

Definition at line 45 of file date_datetime.h.

5.2.2.8 DateVector

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

Definition at line 44 of file date_datetime.h.

5.2.2.9 DottedPair

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

Definition at line 45 of file DottedPair.h.

5.2.2.10 DoubleVector

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

Definition at line 31 of file instantiation.h.

5.2.2.11 Environment

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

Definition at line 403 of file Environment.h.

5.2.2.12 ExpressionMatrix

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

Definition at line 50 of file instantiation.h.

5.2.2.13 ExpressionVector

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

Definition at line 38 of file instantiation.h.

5.2.2.14 Formula

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

Definition at line 77 of file Formula.h.

5.2.2.15 Function

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

Examples

SugarPerformance/Timer.h.

Definition at line 124 of file Function.h.

5.2.2.16 GenericMatrix

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

Definition at line 48 of file instantiation.h.

5.2.2.17 GenericVector

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

Definition at line 36 of file instantiation.h.

5.2.2.18 IntegerMatrix

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

Definition at line 41 of file instantiation.h.

5.2.2.19 IntegerVector

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

Definition at line 28 of file instantiation.h.

5.2.2.20 InternalFunction

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

Definition at line 64 of file InternalFunction.h.

5.2.2.21 Language

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

Definition at line 160 of file Language.h.

5.2.2.22 List

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

Definition at line 37 of file instantiation.h.

5.2.2.23 ListMatrix

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

Definition at line 49 of file instantiation.h.

5.2.2.24 LogicalMatrix

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

Definition at line 42 of file instantiation.h.

5.2.2.25 LogicalVector

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

Definition at line 29 of file instantiation.h.

5.2.2.26 nanotime_t

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

Definition at line 47 of file Timer.h.

5.2.2.27 NumericMatrix

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

Definition at line 43 of file instantiation.h.

5.2.2.28 NumericVector

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

Definition at line 30 of file instantiation.h.

5.2.2.29 Pairlist

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

Definition at line 53 of file Pairlist.h.

5.2.2.30 Promise

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

Definition at line 78 of file Promise.h.

5.2.2.31 RawMatrix

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

Definition at line 44 of file instantiation.h.

5.2.2.32 RawVector

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

Definition at line 32 of file instantiation.h.

5.2.2.33 Reference

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

Definition at line 71 of file Reference.h.

5.2.2.34 RObject

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

Definition at line 58 of file RObject.h.

5.2.2.35 S4

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

Definition at line 77 of file S4.h.

5.2.2.36 StretchyList

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

Definition at line 83 of file StretchyList.h.

5.2.2.37 StringMatrix

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

Definition at line 47 of file instantiation.h.

5.2.2.38 StringVector

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

Definition at line 35 of file instantiation.h.

5.2.2.39 Symbol

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

Definition at line 84 of file Symbol.h.

5.2.2.40 ValidConstructor

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

Definition at line 139 of file Module.h.

5.2.2.41 ValidMethod

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

Definition at line 140 of file Module.h.

5.2.2.42 WeakReference

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

Definition at line 64 of file WeakReference.h.

5.2.3 Function Documentation

5.2.3.1 __any() [1/2]

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

Definition at line 32 of file algo.h.

Referenced by any().

5.2.3.2 __any() [2/2]

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

Definition at line 39 of file algo.h.

5.2.3.3 __any_if() [1/2]

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

Definition at line 98 of file algo.h.

Referenced by any_if().

5.2.3.4 __any_if() [2/2]

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

Definition at line 105 of file algo.h.

5.2.3.5 all()

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

Definition at line 84 of file all.h.

Referenced by ls().

5.2.3.6 any() [1/2]

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

Definition at line 81 of file any.h.

5.2.3.7 any() [2/2]

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

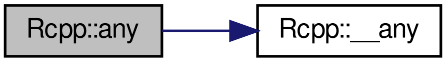
stl like algorithm to identify if any of the objects in the range is equal to the value

Definition at line 89 of file algo.h.

References __any().

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

Here is the call graph for this function:



5.2.3.8 any_if()

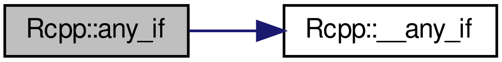
```
template<class InputIterator , class Predicate >
bool Rcpp::any_if (
            InputIterator first,
            InputIterator last,
            Predicate pred )  [inline]
```

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

Definition at line 155 of file algo.h.

References __any_if().

Here is the call graph for this function:



5.2.3.9 as() [1/2]

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

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

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

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

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

Definition at line 151 of file as.h.

Referenced by CppProperty_GetMethod_SetMethod< Class, PROP >::set(), CppProperty_GetConstMethod_SetMethod< Class, PROP >::set(), CppProperty_GetMethod_SetPointer< Class, PROP >::set(), CppProperty_GetConstMethod_SetPointer< Class, PROP >::set(), CppProperty_GetPointer_SetMethod< Class, PROP >::set(), and CppProperty_GetPointer_SetPointer< Class, PROP >::set().

5.2.3.10 as() [2/2]

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

Definition at line 164 of file as.h.

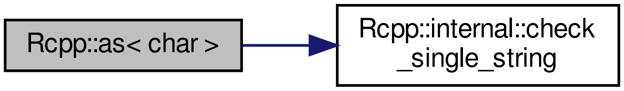
5.2.3.11 as< char >()

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

Definition at line 155 of file as.h.

References Rcpp::internal::check_single_string().

Here is the call graph for this function:



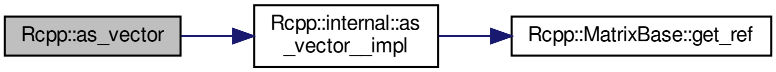
5.2.3.12 as_vector()

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

Definition at line 55 of file as_vector.h.

References Rcpp::internal::as_vector__impl().

Here is the call graph for this function:



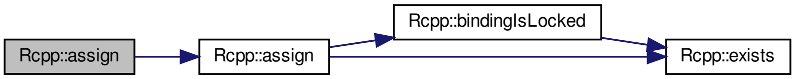
5.2.3.13 assign() [1/3]

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

Definition at line 214 of file Environment.h.

References assign().

Here is the call graph for this function:



5.2.3.14 assign() [2/3]

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

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

Parameters

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

5.2.3.15 assign() [3/3]

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

Attempts to assign x to name in this environment

Parameters

name	name of the object to assign
x	object to assign

Returns

true if the assign was successfull see ?bindingIsLocked

Exceptions

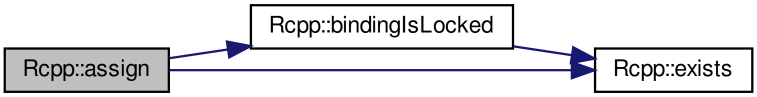
binding_is_locked	if the binding is locked

Definition at line 207 of file Environment.h.

References bindingIsLocked(), and exists().

Referenced by assign().

Here is the call graph for this function:



5.2.3.16 bare_as()

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

Definition at line 160 of file as.h.

5.2.3.17 base_env()

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

Returns

the base environment. See ?baseenv

Definition at line 346 of file Environment.h.

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

5.2.3.18 base_namespace()

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

Returns

the base namespace. See ?baseenv

Definition at line 353 of file Environment.h.

5.2.3.19 bindingIsActive()

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

Parameters

name	name of a binding

Returns

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

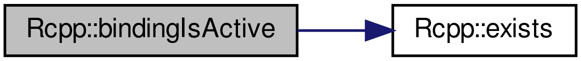
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 315 of file Environment.h.

References exists().

Here is the call graph for this function:



5.2.3.20 bindingIsLocked()

```
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

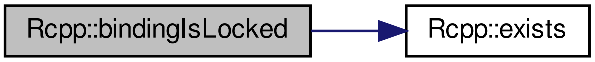
no_such_binding	if there is no such binding in this environment

Definition at line 300 of file Environment.h.

References exists().

Referenced by assign(), and remove().

Here is the call graph for this function:



5.2.3.21 checkUserInterrupt()

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

Definition at line 59 of file Interrupt.h.

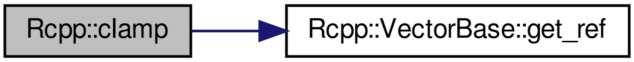
5.2.3.22 clamp()

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

Definition at line 84 of file clamp.h.

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

Here is the call graph for this function:



5.2.3.23 clone()

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

Definition at line 33 of file clone.h.

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

5.2.3.24 col()

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

Definition at line 55 of file col.h.

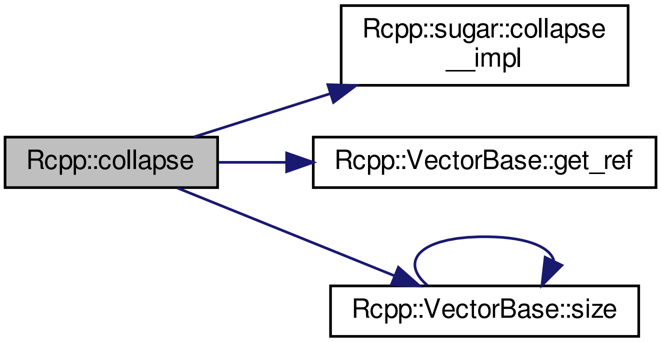
5.2.3.25 collapse()

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

Definition at line 41 of file collapse.h.

References Rcpp::sugar::collapse__impl(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



5.2.3.26 colMeans()

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

Definition at line 951 of file rowSums.h.

5.2.3.27 colnames()

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

Definition at line 213 of file Matrix.h.

5.2.3.28 colSums()

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

Definition at line 933 of file rowSums.h.

5.2.3.29 converter()

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

Definition at line 377 of file Module.h.

5.2.3.30 cummax() [1/2]

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

Definition at line 59 of file cummax.h.

5.2.3.31 cummax() [2/2]

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

Definition at line 64 of file cummax.h.

5.2.3.32 cummin() [1/2]

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

Definition at line 59 of file cummin.h.

5.2.3.33 cummin() [2/2]

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

Definition at line 64 of file cummin.h.

5.2.3.34 cumprod() [1/3]

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

Definition at line 69 of file cumprod.h.

5.2.3.35 cumprod() [2/3]

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

Definition at line 59 of file cumprod.h.

5.2.3.36 cumprod() [3/3]

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

Definition at line 64 of file cumprod.h.

5.2.3.37 cumsum() [1/2]

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

Definition at line 59 of file cumsum.h.

5.2.3.38 cumsum() [2/2]

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

Definition at line 64 of file cumsum.h.

5.2.3.39 detzcode()

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

Definition at line 482 of file date.cpp.

Referenced by tzload().

5.2.3.40 detzcode64()

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

Definition at line 489 of file date.cpp.

Referenced by tzload().

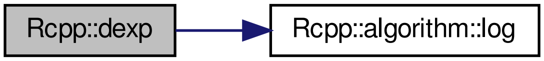
5.2.3.41 dexp()

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

Definition at line 81 of file exp.h.

References Rcpp::algorithm::log().

Here is the call graph for this function:



5.2.3.42 diag()

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

Definition at line 83 of file diag.h.

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

5.2.3.43 diff() [1/2]

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

Definition at line 124 of file diff.h.

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

5.2.3.44 diff() [2/2]

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

Definition at line 131 of file diff.h.

5.2.3.45 differ_by_repeat()

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

Definition at line 496 of file date.cpp.

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

Referenced by tzload().

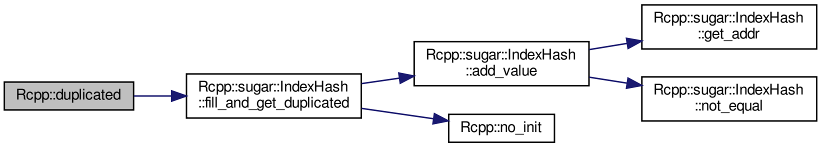
5.2.3.46 duplicated()

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

Definition at line 28 of file duplicated.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated().

Here is the call graph for this function:


Rcpp::duplicated
Rcpp::sugar::IndexHash ::fill_and_get_duplicated
Rcpp::sugar::IndexHash ::add_value
Rcpp::no_init
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

5.2.3.47 empty_env()

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

Returns

The empty environment. See ?emptyenv

Definition at line 339 of file Environment.h.

5.2.3.48 Environment_Impl() [1/3]

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

Gets the environment associated with the given name

Parameters

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

5.2.3.49 Environment_Impl() [2/3]

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

Gets the environment in the given position of the search path

Parameters

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

5.2.3.50 Environment_Impl() [3/3]

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

wraps the given environment

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

Definition at line 57 of file Environment.h.

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

5.2.3.51 eval() [1/2]

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

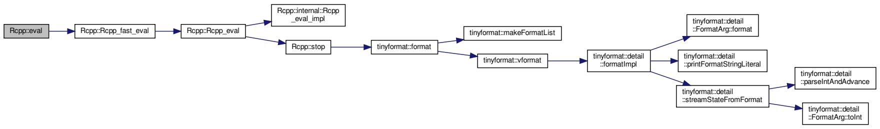
eval this call in the global environment

Definition at line 135 of file Language.h.

References Rcpp_fast_eval().

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

Here is the call graph for this function:


Rcpp::eval
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

5.2.3.52 eval() [2/2]

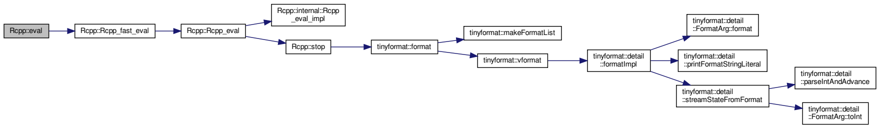
```
SEXP Rcpp::eval (
            SEXP env ) const
```

eval this call in the requested environment

Definition at line 142 of file Language.h.

References Rcpp_fast_eval().

Here is the call graph for this function:


Rcpp::eval
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

5.2.3.53 exists()

```
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 190 of file Environment.h.

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

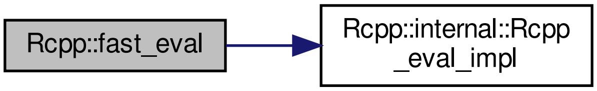
5.2.3.54 fast_eval() [1/2]

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

Definition at line 146 of file Language.h.

References Rcpp::internal::Rcpp_eval_impl().

Here is the call graph for this function:



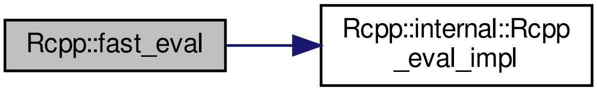
5.2.3.55 fast_eval() [2/2]

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

Definition at line 149 of file Language.h.

References Rcpp::internal::Rcpp_eval_impl().

Here is the call graph for this function:



5.2.3.56 finalizer_wrapper()

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

Definition at line 35 of file XPtr.h.

References DEMANGLE, and RCPP_DEBUG_3.

5.2.3.57 find() [1/2]

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

Get an object from the environment or one of its parents

Parameters

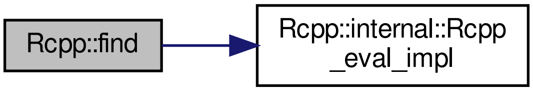
name	name of the object

Definition at line 145 of file Environment.h.

References Rcpp::internal::Rcpp_eval_impl().

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

Here is the call graph for this function:



5.2.3.58 find() [2/2]

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

Get an object from the environment or one of its parents

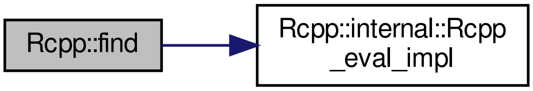
Parameters

name	symbol name to call

Definition at line 165 of file Environment.h.

References Rcpp::internal::Rcpp_eval_impl().

Here is the call graph for this function:



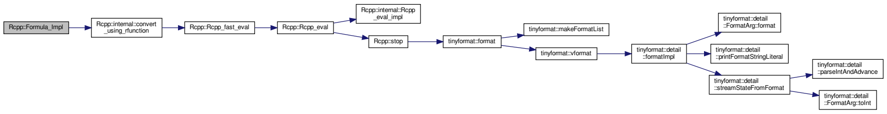
5.2.3.59 Formula_Impl() [1/2]

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

Definition at line 69 of file Formula.h.

References Rcpp::internal::convert_using_rfunction().

Here is the call graph for this function:


Rcpp::Formula_Impl
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

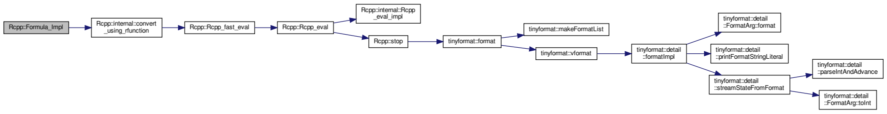
5.2.3.60 Formula_Impl() [2/2]

```
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:


Rcpp::Formula_Impl
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.2.3.61 get() [1/2]

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

Get an object from the environment

Parameters

name	name of the object

Returns

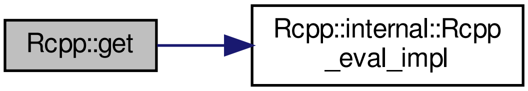
a SEXP (possibly R_NilValue)

Definition at line 103 of file Environment.h.

References Rcpp::internal::Rcpp_eval_impl().

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator SEXP(), Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator SEXP(), Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >↩ ::const_AttributeProxy::operator T(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator T(), Rcpp::↩ BindingPolicy< EnvironmentClass >::const_Binding::operator T(), Rcpp::DottedPairProxyPolicy< CLASS >::Dotted↩ PairProxy::operator T(), Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator T(), Rcpp::Field↩ ProxyPolicy< CLASS >::FieldProxy::operator T(), Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator T(), Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::operator T(), Rcpp::SlotProxyPolicy< CLASS >::Slot↩ Proxy::operator T(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator+=().

Here is the call graph for this function:



5.2.3.62 get() [2/2]

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

Get an object from the environment

Parameters

name	symbol name to call

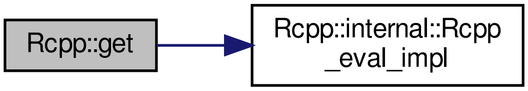
Returns

a SEXP (possibly R_NilValue)

Definition at line 124 of file Environment.h.

References Rcpp::internal::Rcpp_eval_impl().

Here is the call graph for this function:



5.2.3.63 get_return_type()

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

Definition at line 42 of file get_return_type.h.

5.2.3.64 get_return_type< Rcpp::CharacterVector >()

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

Definition at line 74 of file get_return_type.h.

5.2.3.65 get_return_type< Rcpp::ExpressionVector >()

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

Definition at line 66 of file get_return_type.h.

5.2.3.66 get_return_type< Rcpp::IntegerVector >()

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

Definition at line 54 of file get_return_type.h.

5.2.3.67 get_return_type< Rcpp::List >()

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

Definition at line 70 of file get_return_type.h.

5.2.3.68 get_return_type< Rcpp::NumericVector >()

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

Definition at line 58 of file get_return_type.h.

5.2.3.69 get_return_type< Rcpp::RawVector >()

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

Definition at line 62 of file get_return_type.h.

5.2.3.70 get_return_type< SEXP >()

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

Definition at line 50 of file get_return_type.h.

5.2.3.71 get_return_type< void_type >()

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

Definition at line 46 of file get_return_type.h.

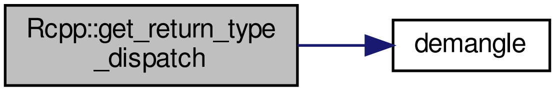
5.2.3.72 get_return_type_dispatch() [1/2]

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

Definition at line 30 of file get_return_type.h.

References demangle().

Here is the call graph for this function:



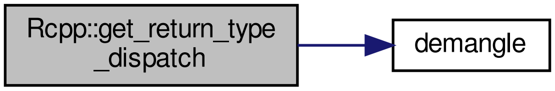
5.2.3.73 get_return_type_dispatch() [2/2]

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

Definition at line 34 of file get_return_type.h.

References demangle().

Here is the call graph for this function:



5.2.3.74 getnum()

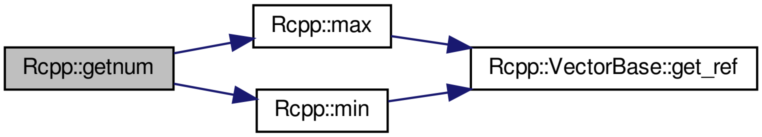
```
static const char * Rcpp::getnum (
              const char * strp,
              int *const nump,
              const int min,
              const int max )  [static]
```

Definition at line 568 of file date.cpp.

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

Referenced by getrule(), and getsecs().

Here is the call graph for this function:



5.2.3.75 getoffset()

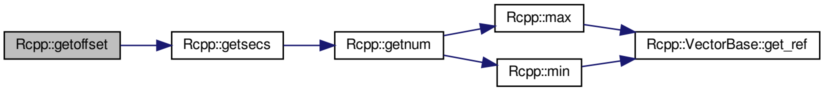
```
static const char* Rcpp::getoffset (
              const char * strp,
              int_fast32_t *const offsetp )  [static]
```

Definition at line 521 of file date.cpp.

References getsecs().

Referenced by tzparse().

Here is the call graph for this function:



5.2.3.76 getqzname()

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

Definition at line 513 of file date.cpp.

Referenced by tzparse().

5.2.3.77 getrule()

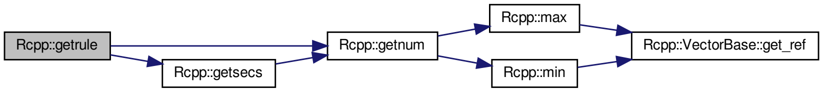
```
static const char * Rcpp::getrule (
             const char * strp,
             struct rule *const rulep )  [static]
```

Definition at line 587 of file date.cpp.

References DAY_OF_YEAR, DAYSPERLYEAR, DAYSPERNYEAR, DAYSPERWEEK, getnum(), getsecs(), is_digit, JULIAN_DAY, MONSPERYEAR, MONTH_NTH_DAY_OF_WEEK, Rcpp::rule::r_day, Rcpp::rule::r_mon, Rcpp::rule::r↩ _time, Rcpp::rule::r_type, Rcpp::rule::r_week, and SECSPERHOUR.

Referenced by tzparse().

Here is the call graph for this function:



5.2.3.78 getsecs()

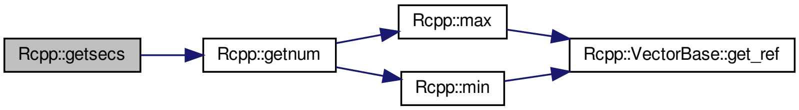
```
static const char * Rcpp::getsecs (
             const char * strp,
             int_fast32_t * secsp )  [static]
```

Definition at line 537 of file date.cpp.

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

Referenced by getoffset(), and getrule().

Here is the call graph for this function:



5.2.3.79 getzname()

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

Definition at line 504 of file date.cpp.

References is_digit.

Referenced by tzparse().

5.2.3.80 global_env()

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

Returns

the global environment. See ?globalenv

Definition at line 332 of file Environment.h.

5.2.3.81 gmtime_()

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

Definition at line 136 of file routines.h.

References GET_CALLABLE, and tm.

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

5.2.3.82 gmtload()

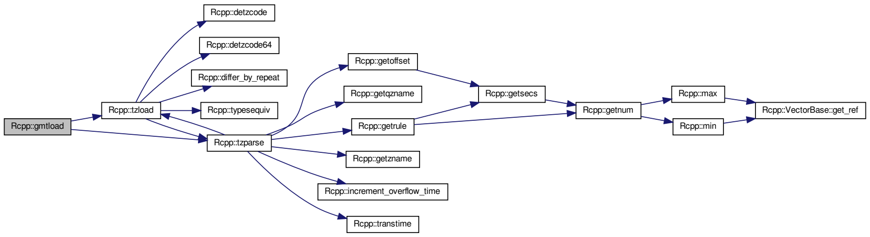
```
static void Rcpp::gmtload (
             struct state *const sp )  [static]
```

Definition at line 1356 of file date.cpp.

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

Referenced by gmtsub().

Here is the call graph for this function:



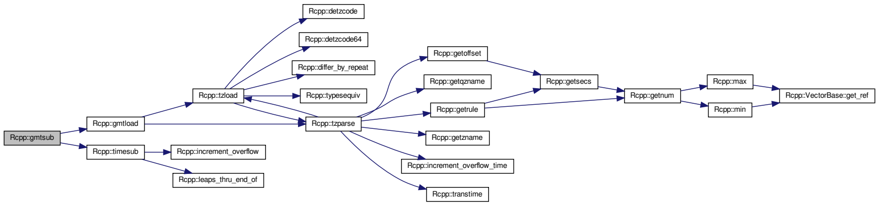
5.2.3.83 gmtsub()

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

Definition at line 1365 of file date.cpp.

References gmt_is_set, gmtload(), gmtptr, timesub(), and tm.

Here is the call graph for this function:



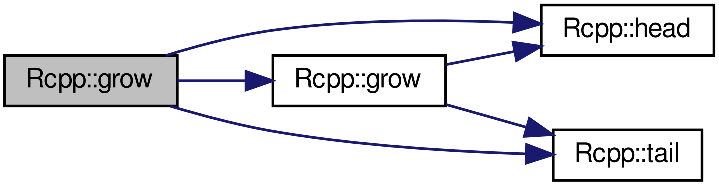
5.2.3.84 grow() [1/3]

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

Definition at line 68 of file grow.h.

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

Here is the call graph for this function:



5.2.3.85 grow() [2/3]

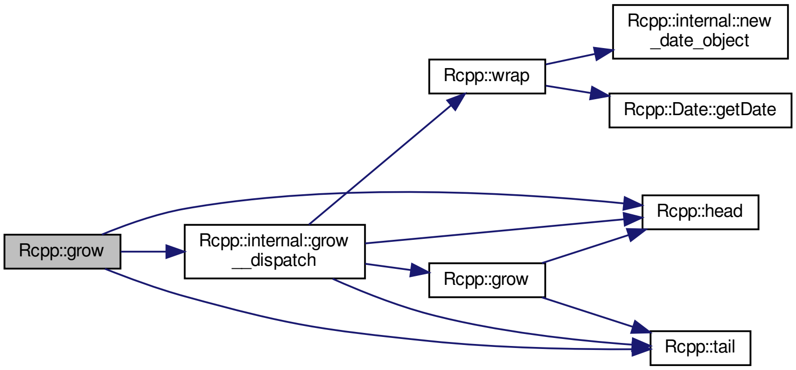
```
template<typename T >
SEXP Rcpp::grow (
            const T & head,
            SEXP tail )
```

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

Definition at line 63 of file grow.h.

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

Here is the call graph for this function:



5.2.3.86 grow() [3/3]

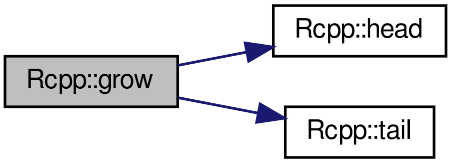
```
SEXP Rcpp::grow (
            SEXP head,
            SEXP tail )  [inline]
```

Definition at line 34 of file grow.h.

References head(), and tail().

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

Here is the call graph for this function:



5.2.3.87 head()

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

Definition at line 53 of file head.h.

Referenced by grow(), and Rcpp::internal::grow__dispatch().

5.2.3.88 ifelse() [1/7]

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

Definition at line 419 of file ifelse.h.

5.2.3.89 ifelse() [2/7]

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

Definition at line 342 of file ifelse.h.

5.2.3.90 ifelse() [3/7]

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

Definition at line 371 of file ifelse.h.

5.2.3.91 ifelse() [4/7]

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

Definition at line 383 of file ifelse.h.

5.2.3.92 ifelse() [5/7]

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

Definition at line 395 of file ifelse.h.

5.2.3.93 ifelse() [6/7]

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

Definition at line 407 of file ifelse.h.

5.2.3.94 ifelse() [7/7]

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

Definition at line 357 of file ifelse.h.

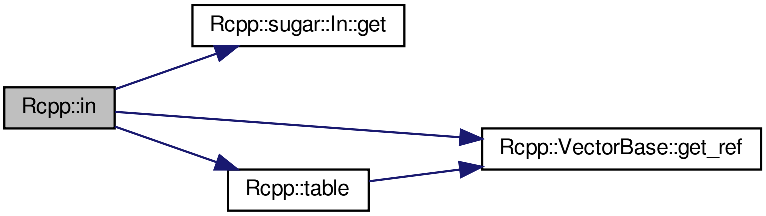
5.2.3.95 in()

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

Definition at line 77 of file unique.h.

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

Here is the call graph for this function:



5.2.3.96 increment_overflow()

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

Definition at line 453 of file date.cpp.

Referenced by timesub().

5.2.3.97 increment_overflow_time()

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

Definition at line 468 of file date.cpp.

References time_t_max, time_t_min, and TYPE_SIGNED.

Referenced by tzparse().

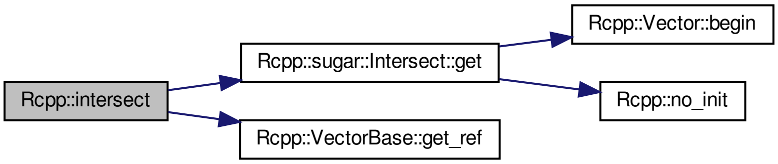
5.2.3.98 intersect()

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

Definition at line 171 of file setdiff.h.

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

Here is the call graph for this function:



5.2.3.99 is()

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

identify if an x can be seen as the T type

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

Definition at line 53 of file is.h.

Referenced by RCPP_API_CLASS().

5.2.3.100 is_false()

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

Definition at line 33 of file is.h.

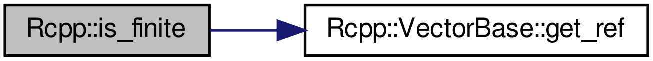
5.2.3.101 is_finite()

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

Definition at line 49 of file is_finite.h.

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

Here is the call graph for this function:



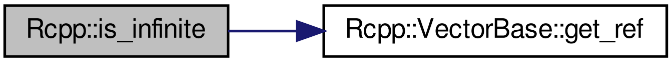
5.2.3.102 is_infinite()

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

Definition at line 49 of file is_infinite.h.

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

Here is the call graph for this function:



5.2.3.103 is_na() [1/6]

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

Definition at line 95 of file is_na.h.

5.2.3.104 is_na() [2/6]

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

Definition at line 99 of file is_na.h.

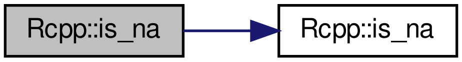
5.2.3.105 is_na() [3/6]

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

Definition at line 38 of file is.h.

References is_na().

Here is the call graph for this function:



5.2.3.106 is_na() [4/6]

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

Definition at line 91 of file is_na.h.

Referenced by is_na(), Rcpp::sugar::na_omit_impl(), Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::operator result_type(), Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type(), Rcpp::algorithm::helpers::log↩ ::operator()(), Rcpp::algorithm::helpers::exp::operator()(), and Rcpp::algorithm::helpers::sqrt::operator()().

5.2.3.107 is_na() [5/6]

```
sugar::IsNa_Vector_is_na<NumericVector> Rcpp::is_na (
            newDatetimeVector & x )  [inline]
```

Definition at line 103 of file is_na.h.

5.2.3.108 is_na() [6/6]

```
sugar::IsNa_Vector_is_na<NumericVector> Rcpp::is_na (
            newDateVector & x )  [inline]
```

Definition at line 106 of file is_na.h.

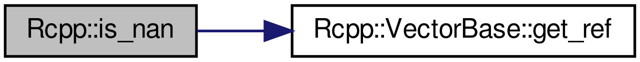
5.2.3.109 is_nan()

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

Definition at line 49 of file is_nan.h.

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

Here is the call graph for this function:



5.2.3.110 is_true()

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

Definition at line 28 of file is.h.

5.2.3.111 is_user_database()

```
bool Rcpp::is_user_database ( ) const
```

Indicates if this is a user defined database.

Definition at line 324 of file Environment.h.

Referenced by ls().

5.2.3.112 isLocked()

```
bool Rcpp::isLocked ( ) const
```

Returns

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

Definition at line 233 of file Environment.h.

5.2.3.113 Language_Impl() [1/5]

```
Rcpp::Language_Impl ( )
```

Definition at line 42 of file Language.h.

5.2.3.114 Language_Impl() [2/5]

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

Creates a call to the function

Parameters

function	function to call

Definition at line 84 of file Language.h.

5.2.3.115 Language_Impl() [3/5]

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

Creates a call using the given symbol as the function name

Parameters

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 63 of file Language.h.

5.2.3.116 Language_Impl() [4/5]

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

Creates a call using the given symbol as the function name

Parameters

symbol	symbol name to call

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

> call("rnorm")

Definition at line 75 of file Language.h.

5.2.3.117 Language_Impl() [5/5]

```
Rcpp::Language_Impl (
             SEXP x )
```

Attempts to convert the SEXP to a call

Exceptions

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

Definition at line 50 of file Language.h.

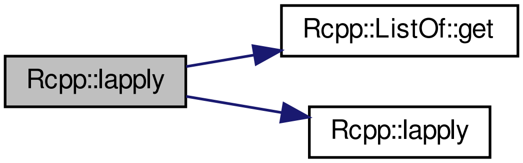
5.2.3.118 lapply() [1/2]

```
template<typename T , typename Function >
List Rcpp::lapply (
             const ListOf< T > & t,
             Function fun )
```

Definition at line 129 of file ListOf.h.

References Rcpp::ListOf< T >::get(), and lapply().

Here is the call graph for this function:



5.2.3.119 lapply() [2/2]

```
template<int RTYPE, bool NA, typename T , typename Function >
sugar::Lapply<RTYPE,NA,T,Function> Rcpp::lapply (
             const Rcpp::VectorBase< RTYPE, NA, T > & t,
             Function fun )  [inline]
```

Definition at line 55 of file lapply.h.

Referenced by lapply().

5.2.3.120 leaps_thru_end_of()

```
static int Rcpp::leaps_thru_end_of (
             const int y )  [static]
```

Definition at line 1222 of file date.cpp.

Referenced by timesub().

5.2.3.121 lock()

```
void Rcpp::lock (
             bool bindings = false )
```

locks this environment. See ?lockEnvironment

Parameters

bindings	also lock the bindings of this environment ?

Definition at line 264 of file Environment.h.

5.2.3.122 lockBinding()

```
void Rcpp::lockBinding (
             const std::string & name )
```

Locks the given binding in the environment. see ?bindingIsLocked

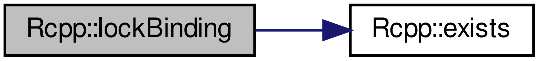
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 274 of file Environment.h.

References exists().

Here is the call graph for this function:



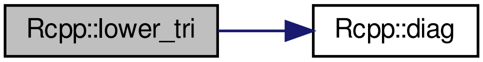
5.2.3.123 lower_tri()

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

Definition at line 63 of file lower_tri.h.

References diag().

Here is the call graph for this function:



5.2.3.124 ls()

```
SEXP Rcpp::ls (
            bool all ) const
```

The list of objects in the environment

the same as calling this from R:

> ls(envir = this, all = all)

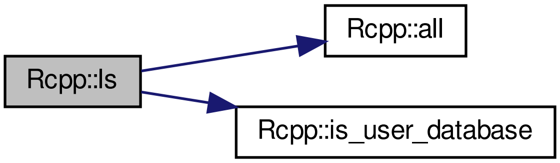
Parameters

all	same meaning as in ?ls

Definition at line 85 of file Environment.h.

References all(), and is_user_database().

Here is the call graph for this function:



5.2.3.125 make_string_transformer()

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

Definition at line 50 of file StringTransformer.h.

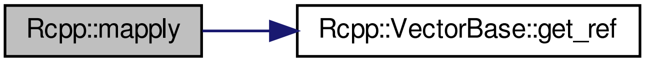
5.2.3.126 mapply() [1/4]

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
sugar::Mapply_2<RTYPE,NA_1,T_1,NA_2,T_2,Function> Rcpp::mapply (
            const Rcpp::VectorBase< RTYPE, NA_1, T_1 > & t1,
            const Rcpp::VectorBase< RTYPE, NA_2, T_2 > & t2,
            Function fun ) [inline]
```

Definition at line 126 of file mapply_2.h.

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

Here is the call graph for this function:



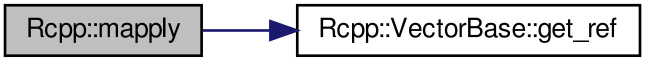
5.2.3.127 mapply() [2/4]

```
template<int RTYPE, bool NA_1, typename T_1 , typename Function >
sugar::Mapply_2_Vector_Primitive<RTYPE,NA_1,T_1,double,Function> Rcpp::mapply (
             const Rcpp::VectorBase< RTYPE, NA_1, T_1 > & t1,
             double t2,
             Function fun )  [inline]
```

Definition at line 132 of file mapply_2.h.

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

Here is the call graph for this function:



5.2.3.128 mapply() [3/4]

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
sugar::Mapply_3<RTYPE_1,NA_1,T_1,RTYPE_2,NA_2,T_2,RTYPE_3, NA_3,T_3,Function> Rcpp::mapply (
             const Rcpp::VectorBase< RTYPE_1, NA_1, T_1 > & t1,
             const Rcpp::VectorBase< RTYPE_2, NA_2, T_2 > & t2,
             const Rcpp::VectorBase< RTYPE_3, NA_3, T_3 > & t3,
             Function fun )  [inline]
```

Definition at line 76 of file mapply_3.h.

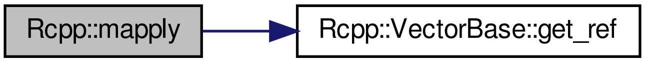
5.2.3.129 mapply() [4/4]

```
template<int RTYPE, bool NA_2, typename T_2 , typename Function >
sugar::Mapply_2_Primitive_Vector<RTYPE,double, NA_2,T_2,Function> Rcpp::mapply (
            double t1,
            const Rcpp::VectorBase< RTYPE, NA_2, T_2 > & t2,
            Function fun )  [inline]
```

Definition at line 138 of file mapply_2.h.

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

Here is the call graph for this function:



5.2.3.130 match()

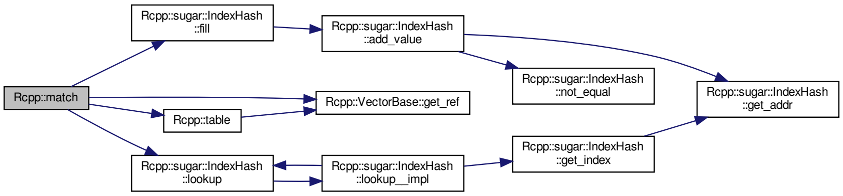
```
template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
IntegerVector Rcpp::match (
            const VectorBase< RTYPE, NA, T > & x,
            const VectorBase< RTYPE, RHS_NA, RHS_T > & table_ )  [inline]
```

Definition at line 28 of file match.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), and table().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::fill
Rcpp::sugar::IndexHash ::add_value
Rcpp::sugar::IndexHash ::not_equal
Rcpp::sugar::IndexHash ::get_addr
Rcpp::match
Rcpp::VectorBase::get_ref
Rcpp::table
Rcpp::sugar::IndexHash ::get_index
Rcpp::sugar::IndexHash ::lookup
Rcpp::sugar::IndexHash ::lookup__impl

5.2.3.131 max()

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

Definition at line 82 of file max.h.

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

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

Here is the call graph for this function:

Rcpp::max
Rcpp::VectorBase::get_ref

5.2.3.132 mean() [1/4]

```
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.

5.2.3.133 mean() [2/4]

```
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.

5.2.3.134 mean() [3/4]

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

Definition at line 155 of file mean.h.

5.2.3.135 mean() [4/4]

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

Definition at line 140 of file mean.h.

Referenced by Rcpp::sugar::Var< RTYPE, NA, T >::get(), Rcpp::sugar::Var< CPLXSXP, NA, T >::get(), and rnorm().

5.2.3.136 median()

```
template<int RTYPE, bool NA, typename T >
sugar::median_detail::result<RTYPE>::type Rcpp::median (
             const Rcpp::VectorBase< RTYPE, NA, T > & x,
             bool na_rm = false )  [inline]
```

Definition at line 283 of file median.h.

5.2.3.137 message()

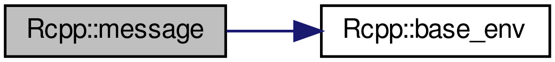
```
void Rcpp::message (
             SEXP s )  [inline]
```

Definition at line 26 of file message.h.

References base_env().

Referenced by stop(), and warning().

Here is the call graph for this function:



5.2.3.138 min()

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

Definition at line 82 of file min.h.

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

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

Here is the call graph for this function:

5.2.3.139 mktime00()

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

Definition at line 130 of file routines.h.

References GET_CALLABLE, and tm.

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

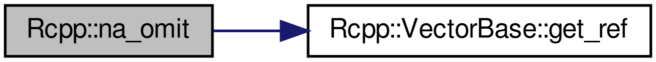
5.2.3.140 na_omit()

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

Definition at line 75 of file na_omit.h.

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

Here is the call graph for this function:



5.2.3.141 Named() [1/2]

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

Definition at line 40 of file Named.h.

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

5.2.3.142 Named() [2/2]

```
template<typename T >
traits::named_object<T> Rcpp::Named (
             const std::string & name,
             const T & o )  [inline]
```

Definition at line 44 of file Named.h.

5.2.3.143 namespace_env()

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

Parameters

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

Returns

the namespace of the package

Exceptions

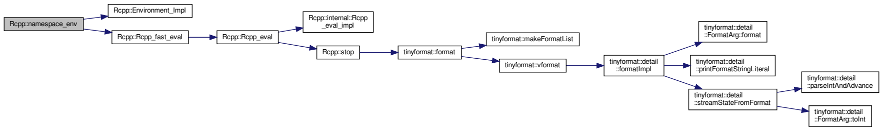
no_such_namespace	

Definition at line 371 of file Environment.h.

References Environment_Impl(), and Rcpp_fast_eval().

Referenced by Rcpp::attributes::checkRSignature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::namespace_env
Rcpp::Environment_Impl
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.2.3.144 new_child()

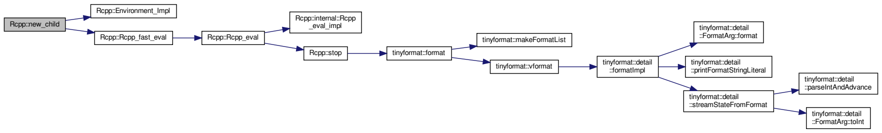
```
Environment_Impl Rcpp::new_child (
             bool hashed ) const
```

creates a new environment whose this is the parent

Definition at line 394 of file Environment.h.

References Environment_Impl(), and Rcpp_fast_eval().

Here is the call graph for this function:


Rcpp::new_child
Rcpp::Environment_Impl
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.2.3.145 new_env() [1/2]

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

Definition at line 47 of file Environment.h.

References _.

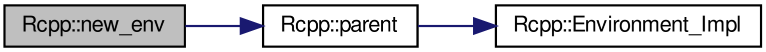
5.2.3.146 new_env() [2/2]

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

Definition at line 52 of file Environment.h.

References _, and parent().

Here is the call graph for this function:



5.2.3.147 no_init() [1/2]

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

Definition at line 81 of file no_init.h.

5.2.3.148 no_init() [2/2]

```
no_init_vector Rcpp::no_init (
            R_xlen_t size ) [inline]
```

Definition at line 77 of file no_init.h.

Referenced by Rcpp::sugar::EmpiricalSample(), Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated(), Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match(), Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec(), Rcpp::sugar::IndexHash< RTYPE >::keys(), Rcpp::sugar::na_omit_impl(), Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), trimws(), and Rcpp::sugar::WalkerSample().

5.2.3.149 noNA() [1/3]

```
template<int RTYPE, bool NA, typename VECTOR >
sugar::Nona<RTYPE,NA,VECTOR> Rcpp::noNA (
            const Rcpp::VectorBase< RTYPE, NA, VECTOR > & vec ) [inline]
```

Definition at line 74 of file nona.h.

Referenced by convolve11cpp().

5.2.3.150 noNA() [2/3]

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

Definition at line 78 of file nona.h.

5.2.3.151 noNA() [3/3]

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

Definition at line 81 of file nona.h.

5.2.3.152 operator SEXP()

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

Definition at line 53 of file StretchyList.h.

5.2.3.153 operator"!=() [1/4]

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

Definition at line 170 of file Date.h.

5.2.3.154 operator"!=() [2/4]

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

Definition at line 164 of file Datetime.h.

5.2.3.155 operator"!=() [3/4]

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

Definition at line 731 of file String.h.

5.2.3.156 operator"!=() [4/4]

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

Definition at line 723 of file String.h.

5.2.3.157 operator*() [1/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive_nona< RTYPE ,
NA , T > >::type Rcpp::operator* (
             const typename sugar::NonaPrimitive< U > & rhs,
             const VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 459 of file times.h.

5.2.3.158 operator*() [2/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive< RTYPE , NA ,
T > >::type Rcpp::operator* (
             const U & rhs,
             const VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 439 of file times.h.

5.2.3.159 operator*() [3/5]

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

Definition at line 473 of file times.h.

5.2.3.160 operator*() [4/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive_nona<RTYPE,NA,T>
>::type Rcpp::operator* (
             const VectorBase< RTYPE, NA, T > & lhs,
             const typename sugar::NonaPrimitive< U > & rhs )  [inline]
```

Definition at line 450 of file times.h.

5.2.3.161 operator*() [5/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive<RTYPE,NA,T>
>::type Rcpp::operator* (
             const VectorBase< RTYPE, NA, T > & lhs,
             const U & rhs )  [inline]
```

Definition at line 429 of file times.h.

5.2.3.162 operator+() [1/8]

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

Definition at line 156 of file Date.h.

5.2.3.163 operator+() [2/8]

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

Definition at line 138 of file Datetime.h.

5.2.3.164 operator+() [3/8]

```
Datetime Rcpp::operator+ (
             const Datetime & datetime,
             int offset )  [inline]
```

Definition at line 148 of file Datetime.h.

5.2.3.165 operator+() [4/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Plus_Vector_Primitive_nona< RTYPE ,
NA , T> >::type Rcpp::operator+ (
             const typename sugar::NonaPrimitive< U > & rhs,
             const VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 478 of file plus.h.

5.2.3.166 operator+() [5/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, typename sugar::Plus_Vector_Primitive< RTYPE
, NA , T> >::type Rcpp::operator+ (
             const U & rhs,
             const VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 458 of file plus.h.

5.2.3.167 operator+() [6/8]

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

Definition at line 492 of file plus.h.

5.2.3.168 operator+() [7/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Plus_Vector_Primitive_nona<RTYPE,NA,T>
>::type Rcpp::operator+ (
             const VectorBase< RTYPE, NA, T > & lhs,
             const typename sugar::NonaPrimitive< U > & rhs )  [inline]
```

Definition at line 469 of file plus.h.

5.2.3.169 operator+() [8/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, typename sugar::Plus_Vector_Primitive<RTYPE,NA,T>
>::type Rcpp::operator+ (
             const VectorBase< RTYPE, NA, T > & lhs,
             const U & rhs )  [inline]
```

Definition at line 448 of file plus.h.

5.2.3.170 operator-() [1/5]

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

Definition at line 164 of file Date.h.

5.2.3.171 operator-() [2/5]

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

Definition at line 158 of file Datetime.h.

5.2.3.172 operator-() [3/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Minus_Primitive_Vector< RTYPE , NA,T>
>::type Rcpp::operator- (
             const U & lhs,
             const VectorBase< RTYPE, NA, T > & rhs )  [inline]
```

Definition at line 427 of file minus.h.

5.2.3.173 operator-() [4/5]

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

Definition at line 440 of file minus.h.

5.2.3.174 operator-() [5/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Minus_Vector_Primitive< RTYPE , NA, T
> >::type Rcpp::operator- (
            const VectorBase< RTYPE, NA, T > & lhs,
            const U & rhs ) [inline]
```

Definition at line 417 of file minus.h.

5.2.3.175 operator/() [1/3]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Divides_Primitive_Vector< RTYPE ,
NA,T> >::type Rcpp::operator/ (
            const U & lhs,
            const VectorBase< RTYPE, NA, T > & rhs ) [inline]
```

Definition at line 423 of file divides.h.

5.2.3.176 operator/() [2/3]

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

Definition at line 436 of file divides.h.

5.2.3.177 operator/() [3/3]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Divides_Vector_Primitive< RTYPE , NA,
T > >::type Rcpp::operator/ (
             const VectorBase< RTYPE, NA, T > & lhs,
             const U & rhs )  [inline]
```

Definition at line 413 of file divides.h.

5.2.3.178 operator<() [1/2]

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

Definition at line 165 of file Date.h.

5.2.3.179 operator<() [2/2]

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

Definition at line 159 of file Datetime.h.

5.2.3.180 operator<<() [1/10]

```
std::ostream& Rcpp::operator<< (
             std::ostream & os,
             const Date d )  [inline]
```

Definition at line 172 of file Date.h.

5.2.3.181 operator<<() [2/10]

```
std::ostream& Rcpp::operator<< (
             std::ostream & os,
             const Datetime d )  [inline]
```

Definition at line 166 of file Datetime.h.

5.2.3.182 operator<<() [3/10]

```
std::ostream& Rcpp::operator<< (
             std::ostream & os,
             const newDatetimeVector d )  [inline]
```

Definition at line 81 of file newDatetimeVector.h.

5.2.3.183 operator<<() [4/10]

```
std::ostream& Rcpp::operator<< (
             std::ostream & os,
             const newDateVector d )  [inline]
```

Definition at line 65 of file newDateVector.h.

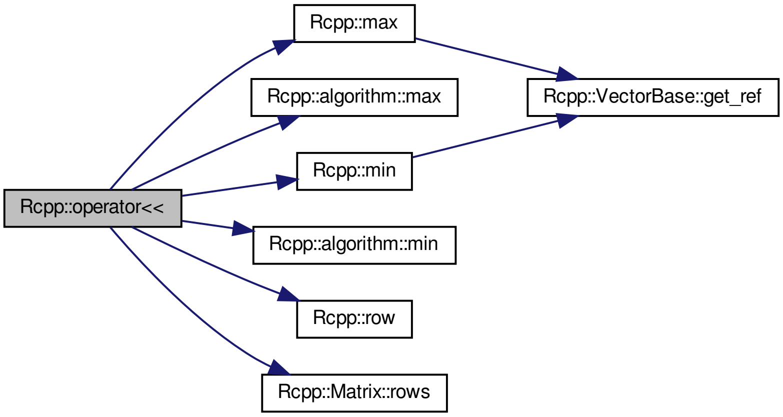
5.2.3.184 operator<<() [5/10]

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

Definition at line 286 of file Matrix.h.

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

Here is the call graph for this function:



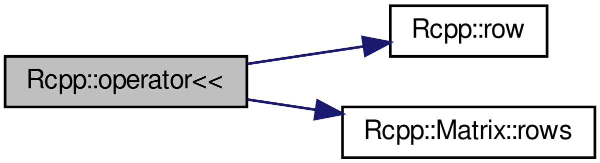
5.2.3.185 operator<<() [6/10]

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

Definition at line 218 of file Matrix.h.

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

Here is the call graph for this function:



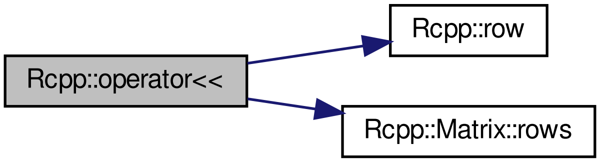
5.2.3.186 operator<<() [7/10]

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

Definition at line 379 of file Matrix.h.

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

Here is the call graph for this function:



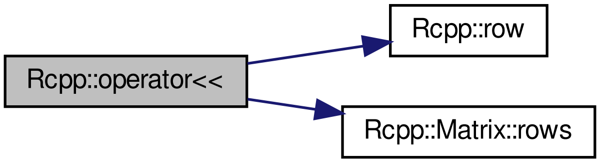
5.2.3.187 operator<<() [8/10]

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

Definition at line 352 of file Matrix.h.

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

Here is the call graph for this function:



5.2.3.188 operator<<() [9/10]

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

Definition at line 1142 of file Vector.h.

5.2.3.189 operator<<() [10/10]

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

Definition at line 1161 of file Vector.h.

5.2.3.190 operator<=() [1/2]

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

Definition at line 169 of file Date.h.

5.2.3.191 operator<=() [2/2]

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

Definition at line 163 of file Datetime.h.

5.2.3.192 operator==() [1/4]

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

Definition at line 167 of file Date.h.

Referenced by RCPP_API_CLASS().

5.2.3.193 operator==() [2/4]

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

Definition at line 161 of file Datetime.h.

5.2.3.194 operator==() [3/4]

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

Definition at line 727 of file String.h.

5.2.3.195 operator==() [4/4]

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

Definition at line 719 of file String.h.

5.2.3.196 operator>() [1/2]

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

Definition at line 166 of file Date.h.

5.2.3.197 operator>() [2/2]

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

Definition at line 160 of file Datetime.h.

5.2.3.198 operator>=() [1/2]

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

Definition at line 168 of file Date.h.

5.2.3.199 operator>=() [2/2]

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

Definition at line 162 of file Datetime.h.

5.2.3.200 outer()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
sugar::Outer<RTYPE,LHS_NA,LHS_T,RHS_NA,RHS_T,Function> Rcpp::outer (
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs,
             Function fun )  [inline]
```

Definition at line 80 of file outer.h.

5.2.3.201 pairlist()

```
SEXP Rcpp::pairlist ( )  [inline]
```

Definition at line 30 of file grow.h.

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

5.2.3.202 Pairlist_Impl() [1/2]

```
Rcpp::Pairlist_Impl ( )
```

Definition at line 41 of file Pairlist.h.

5.2.3.203 Pairlist_Impl() [2/2]

```
Rcpp::Pairlist_Impl (
             SEXP x )
```

Definition at line 42 of file Pairlist.h.

5.2.3.204 parent()

```
Environment_Impl Rcpp::parent ( ) const
```

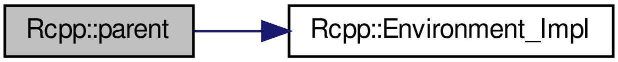
The parent environment of this environment

Definition at line 387 of file Environment.h.

References Environment_Impl().

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn(), class_< Class >::derives(), Rcpp::↩ MatrixColumn< RTYPE >::MatrixColumn(), and new_env().

Here is the call graph for this function:



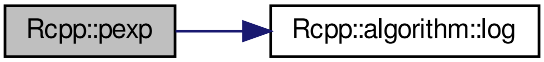
5.2.3.205 pexp()

```
template<bool NA, typename T >
stats::P1<REALSXP,NA,T> Rcpp::pexp (
             const Rcpp::VectorBase< REALSXP, NA, T > & x,
             double shape,
             bool lower = true,
             bool log = false )  [inline]
```

Definition at line 86 of file exp.h.

References Rcpp::algorithm::log().

Here is the call graph for this function:



5.2.3.206 pmax() [1/3]

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

Definition at line 158 of file pmax.h.

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

Here is the call graph for this function:

Rcpp::pmax
Rcpp::VectorBase::get_ref

5.2.3.207 pmax() [2/3]

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
sugar::Pmax_Vector_Primitive<RTYPE,LHS_NA,LHS_T> Rcpp::pmax (
            const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
            typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 170 of file pmax.h.

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

Here is the call graph for this function:

Rcpp::pmax
Rcpp::VectorBase::get_ref

5.2.3.208 pmax() [3/3]

```
template<int RTYPE, bool RHS_NA, typename RHS_T >
sugar::Pmax_Vector_Primitive<RTYPE,RHS_NA,RHS_T> Rcpp::pmax (
            typename Rcpp::traits::storage_type< RTYPE >::type lhs,
            const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 183 of file pmax.h.

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

Here is the call graph for this function:

Rcpp::pmax
Rcpp::VectorBase::get_ref

5.2.3.209 pmin() [1/3]

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

Definition at line 156 of file pmin.h.

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

Here is the call graph for this function:

Rcpp::pmin
Rcpp::VectorBase::get_ref

5.2.3.210 pmin() [2/3]

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

Definition at line 168 of file pmin.h.

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

Here is the call graph for this function:

Rcpp::pmin
Rcpp::VectorBase::get_ref

5.2.3.211 pmin() [3/3]

```
template<int RTYPE, bool RHS_NA, typename RHS_T >
sugar::Pmin_Vector_Primitive<RTYPE,RHS_NA,RHS_T> Rcpp::pmin (
             typename Rcpp::traits::storage_type< RTYPE >::type lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 181 of file pmin.h.

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

Here is the call graph for this function:

Rcpp::pmin
Rcpp::VectorBase::get_ref

5.2.3.212 pow()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
sugar::Pow<RTYPE,NA,T,EXPONENT_TYPE> Rcpp::pow (
             const VectorBase< RTYPE, NA, T > & t,
             EXPONENT_TYPE exponent )  [inline]
```

Definition at line 79 of file pow.h.

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

Referenced by Rcpp::stats::dweibull_1(), Rcpp::sugar::Var< RTYPE, NA, T >::get(), Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::operator[](), Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::operator[](), Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::operator[](), Rcpp::stats::pweibull_1(), and Rcpp::stats↩ ::qweibull_1().

Here is the call graph for this function:

Rcpp::pow
Rcpp::VectorBase::get_ref

5.2.3.213 print()

```
void Rcpp::print (
             SEXP s )  [inline]
```

Definition at line 25 of file print.h.

5.2.3.214 push_back()

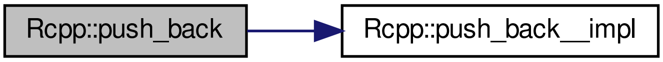
```
template<typename T >
StretchyList_Impl& Rcpp::push_back (
             const T & obj )  [inline]
```

Definition at line 58 of file StretchyList.h.

References push_back__impl().

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

Here is the call graph for this function:



5.2.3.215 push_back__impl() [1/2]

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

5.2.3.216 push_back__impl() [2/2]

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

Referenced by push_back().

5.2.3.217 push_front()

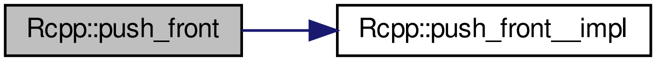
```
template<typename T >
StretchyList_Impl& Rcpp::push_front (
             const T & obj )  [inline]
```

Definition at line 63 of file StretchyList.h.

References push_front__impl().

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

Here is the call graph for this function:



5.2.3.218 push_front__impl() [1/2]

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

5.2.3.219 push_front__impl() [2/2]

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

Referenced by push_front().

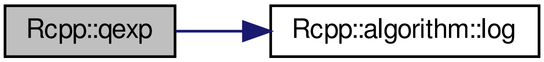
5.2.3.220 qexp()

```
template<bool NA, typename T >
stats::Q1<REALSXP,NA,T> Rcpp::qexp (
             const Rcpp::VectorBase< REALSXP, NA, T > & x,
             double shape,
             bool lower = true,
             bool log = false )  [inline]
```

Definition at line 91 of file exp.h.

References Rcpp::algorithm::log().

Here is the call graph for this function:



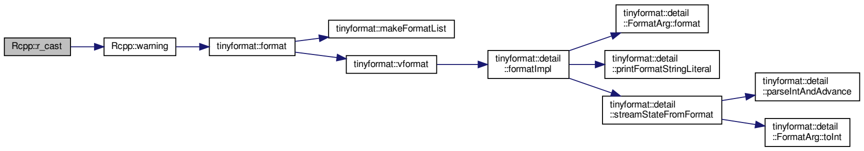
5.2.3.221 r_cast()

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

Definition at line 164 of file r_cast.h.

References warning().

Here is the call graph for this function:



5.2.3.222 range()

```
template<int RTYPE, bool NA, typename T >
sugar::Range<RTYPE,NA,T> Rcpp::range (
            const VectorBase< RTYPE, NA, T > & x )
```

Definition at line 86 of file range.h.

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

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ]().

Here is the call graph for this function:

Rcpp::range
Rcpp::VectorBase::get_ref

5.2.3.223 rbeta()

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

Definition at line 100 of file random.h.

5.2.3.224 rbinom()

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

Definition at line 104 of file random.h.

5.2.3.225 rcauchy() [1/3]

```
NumericVector Rcpp::rcauchy (
             int n )  [inline]
```

Definition at line 128 of file random.h.

5.2.3.226 rcauchy() [2/3]

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

Definition at line 118 of file random.h.

5.2.3.227 rcauchy() [3/3]

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

Definition at line 108 of file random.h.

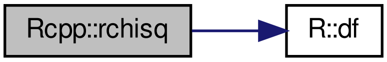
5.2.3.228 rchisq()

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

Definition at line 132 of file random.h.

References R::df().

Here is the call graph for this function:



5.2.3.229 RCPP_API_CLASS() [1/14]

```
Rcpp::RCPP_API_CLASS (
             DottedPair_Impl  )
```

5.2.3.230 RCPP_API_CLASS() [2/14]

```
Rcpp::RCPP_API_CLASS (
             Environment_Impl  )
```

5.2.3.231 RCPP_API_CLASS() [3/14]

```
Rcpp::RCPP_API_CLASS (
             Formula_Impl  )
```

5.2.3.232 RCPP_API_CLASS() [4/14]

```
Rcpp::RCPP_API_CLASS (
             Function_Impl  )
```

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

Parameters

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

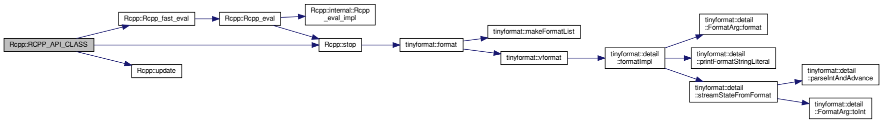
Returns the environment of this function

Returns the body of the function

Definition at line 34 of file Function.h.

References Rcpp_fast_eval(), RCPP_GENERATE_CTOR_ASSIGN, Rcpp_lcons, stop(), and update().

Here is the call graph for this function:


Rcpp::RCPP_API_CLASS
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
Rcpp::update
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

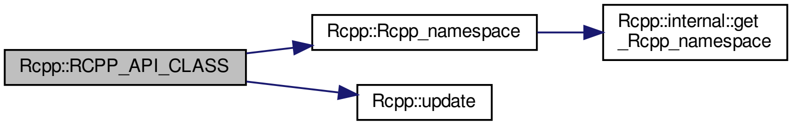
5.2.3.233 RCPP_API_CLASS() [5/14]

```
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:



5.2.3.234 RCPP_API_CLASS() [6/14]

```
Rcpp::RCPP_API_CLASS (
             Language_Impl   )
```

C++ wrapper around calls (LANGSXP SEXP)

This represents calls that can be evaluated

5.2.3.235 RCPP_API_CLASS() [7/14]

```
Rcpp::RCPP_API_CLASS (
             Pairlist_Impl   )
```

5.2.3.236 RCPP_API_CLASS() [8/14]

```
Rcpp::RCPP_API_CLASS (
             Promise_Impl   )
```

Return the result of the PRSEEN macro

Return the result of the PRVALUE macro on the promise

The promise expression: PRCODE

The promise environment : PRENV

Definition at line 27 of file Promise.h.

References RCPP_GENERATE_CTOR_ASSIGN, and update().

Here is the call graph for this function:

Rcpp::RCPP_API_CLASS
Rcpp::update

5.2.3.237 RCPP_API_CLASS() [9/14]

```
Rcpp::RCPP_API_CLASS (
            Reference_Impl  )
```

S4 object (of a reference class)

5.2.3.238 RCPP_API_CLASS() [10/14]

```
Rcpp::RCPP_API_CLASS (
            RObject_Impl  )
```

default constructor. uses R_NilValue

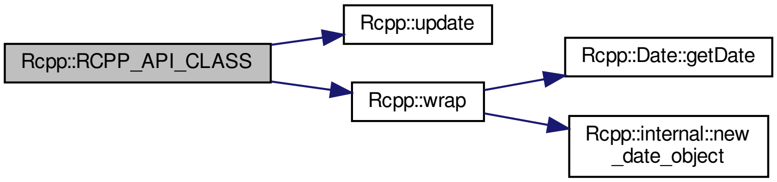
wraps a SEXP. The SEXP is automatically protected from garbage collection by this object and the protection vanishes when this object is destroyed

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:



5.2.3.239 RCPP_API_CLASS() [11/14]

```
Rcpp::RCPP_API_CLASS (
            S4_Impl  )
```

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

Parameters

x	must be an S4 object

Creates an S4 object of the requested class.

Parameters

klass	name of the target S4 class

Exceptions

S4_creation_error	if klass does not map to a known S4 class

Indicates if this object is an instance of the given S4 class

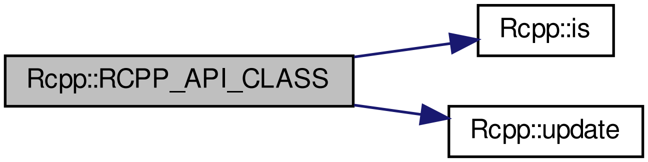
Exceptions

not_s4	if x is not an S4 class

Definition at line 30 of file S4.h.

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

Here is the call graph for this function:



5.2.3.240 RCPP_API_CLASS() [12/14]

```
Rcpp::RCPP_API_CLASS (
             StretchyList_Impl  )
```

StretchyList uses a special pairlist to provide efficient insertion at the front and the end of a pairlist.

This is a C++ abstraction of the functions NewList, GrowList and Insert that are found in places where a pair list has to grow efficiently, e.g. in the R parser (gram.y)

5.2.3.241 RCPP_API_CLASS() [13/14]

```
Rcpp::RCPP_API_CLASS (
            Symbol_Impl )
```

wraps the SEXP into a Symbol object.

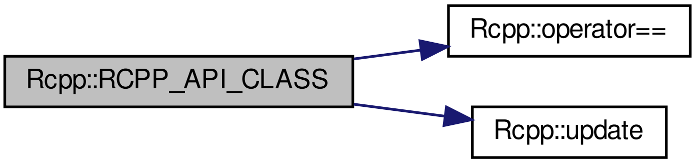
Parameters

x	Accepted SEXP types are SYMSXP, CHARSXP and STRSXP in the last case, the first element of the character vector is silently used

Definition at line 27 of file Symbol.h.

References operator==(), RCPP_GENERATE_CTOR_ASSIGN, and update().

Here is the call graph for this function:



5.2.3.242 RCPP_API_CLASS() [14/14]

```
Rcpp::RCPP_API_CLASS (
            WeakReference_Impl )
```

wraps a weak reference

Parameters

x	presumably a SEXP of SEXTYPE WEAKREFSXP

Exceptions

not_compatible	if x is not a weak reference

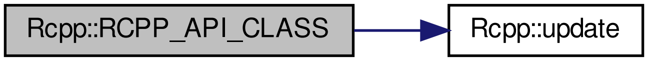
Retrieve the key

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:



5.2.3.243 Rcpp_cerr_get()

```
Rostream< false > & Rcpp::Rcpp_cerr_get ( )  [inline]
```

Definition at line 168 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.2.3.244 Rcpp_cout_get()

```
Rostream< true > & Rcpp::Rcpp_cout_get ( )  [inline]
```

Definition at line 163 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.2.3.245 Rcpp_eval()

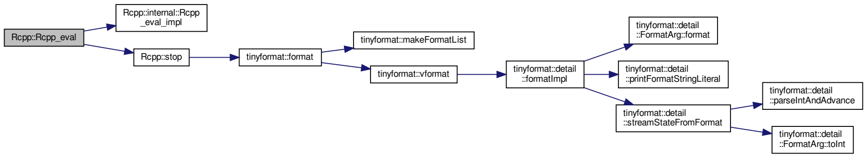
```
SEXP Rcpp::Rcpp_eval (
             SEXP expr,
             SEXP env )  [inline]
```

Definition at line 75 of file Rcpp_eval.h.

References Rcpp::internal::Rcpp_eval_impl(), and stop().

Referenced by Rcpp_fast_eval().

Here is the call graph for this function:


Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.2.3.246 RCPP_EXCEPTION_CLASS()

```
No such field Rcpp::RCPP_EXCEPTION_CLASS (
             reference_creation_error ,
             "Error creating object of reference class"  )
```

5.2.3.247 Rcpp_fast_eval()

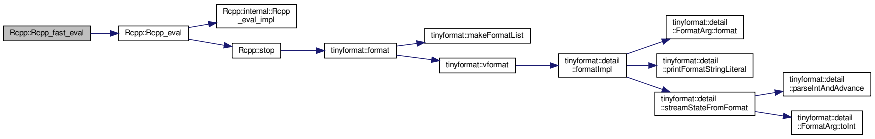
```
SEXP Rcpp::Rcpp_fast_eval (
             SEXP expr,
             SEXP env )  [inline]
```

Definition at line 68 of file Rcpp_eval.h.

References Rcpp_eval().

Referenced by Rcpp::internal::convert_using_rfunction(), eval(), Rcpp::Vector< RTYPE, StoragePolicy >::eval(), Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get(), Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get(), get_last_call(), namespace_env(), new_child(), Rcpp::internal::r_true_cast< STRSXP >(), RCPP_API_CLASS(), remove(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set(), and Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set().

Here is the call graph for this function:


Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

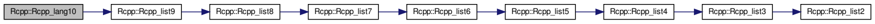
5.2.3.248 Rcpp_lang10()

```
SEXP Rcpp::Rcpp_lang10 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9 )  [inline]
```

Definition at line 127 of file lang.h.

References Rcpp_list9().

Here is the call graph for this function:



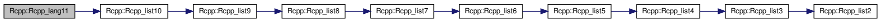
5.2.3.249 Rcpp_lang11()

```
SEXP Rcpp::Rcpp_lang11 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10 )  [inline]
```

Definition at line 143 of file lang.h.

References Rcpp_list10().

Here is the call graph for this function:



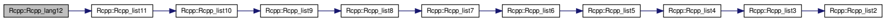
5.2.3.250 Rcpp_lang12()

```
SEXP Rcpp::Rcpp_lang12 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11 )  [inline]
```

Definition at line 159 of file lang.h.

References Rcpp_list11().

Here is the call graph for this function:



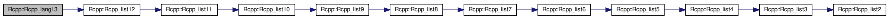
5.2.3.251 Rcpp_lang13()

```
SEXP Rcpp::Rcpp_lang13 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12 )  [inline]
```

Definition at line 175 of file lang.h.

References Rcpp_list12().

Here is the call graph for this function:



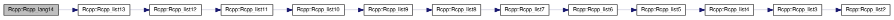
5.2.3.252 Rcpp_lang14()

```
SEXP Rcpp::Rcpp_lang14 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13 )  [inline]
```

Definition at line 191 of file lang.h.

References Rcpp_list13().

Here is the call graph for this function:



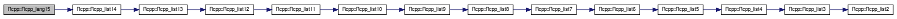
5.2.3.253 Rcpp_lang15()

```
SEXP Rcpp::Rcpp_lang15 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4,
            SEXP x5,
            SEXP x6,
            SEXP x7,
            SEXP x8,
            SEXP x9,
            SEXP x10,
            SEXP x11,
            SEXP x12,
            SEXP x13,
            SEXP x14 )  [inline]
```

Definition at line 207 of file lang.h.

References Rcpp_list14().

Here is the call graph for this function:



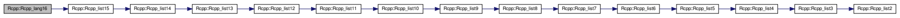
5.2.3.254 Rcpp_lang16()

```
SEXP Rcpp::Rcpp_lang16 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4,
            SEXP x5,
            SEXP x6,
            SEXP x7,
            SEXP x8,
            SEXP x9,
            SEXP x10,
            SEXP x11,
            SEXP x12,
            SEXP x13,
            SEXP x14,
            SEXP x15 )  [inline]
```

Definition at line 223 of file lang.h.

References Rcpp_list15().

Here is the call graph for this function:



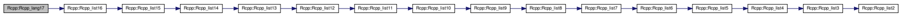
5.2.3.255 Rcpp_lang17()

```
SEXP Rcpp::Rcpp_lang17 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16 )  [inline]
```

Definition at line 239 of file lang.h.

References Rcpp_list16().

Here is the call graph for this function:



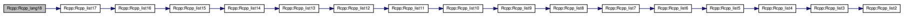
5.2.3.256 Rcpp_lang18()

```
SEXP Rcpp::Rcpp_lang18 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4,
            SEXP x5,
            SEXP x6,
            SEXP x7,
            SEXP x8,
            SEXP x9,
            SEXP x10,
            SEXP x11,
            SEXP x12,
            SEXP x13,
            SEXP x14,
            SEXP x15,
            SEXP x16,
            SEXP x17 )  [inline]
```

Definition at line 255 of file lang.h.

References Rcpp_list17().

Here is the call graph for this function:



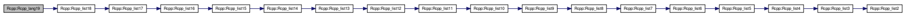
5.2.3.257 Rcpp_lang19()

```
SEXP Rcpp::Rcpp_lang19 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4,
            SEXP x5,
            SEXP x6,
            SEXP x7,
            SEXP x8,
            SEXP x9,
            SEXP x10,
            SEXP x11,
            SEXP x12,
            SEXP x13,
            SEXP x14,
            SEXP x15,
            SEXP x16,
            SEXP x17,
            SEXP x18 )  [inline]
```

Definition at line 271 of file lang.h.

References Rcpp_list18().

Here is the call graph for this function:



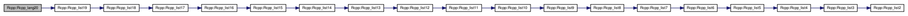
5.2.3.258 Rcpp_lang20()

```
SEXP Rcpp::Rcpp_lang20 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16,
             SEXP x17,
             SEXP x18,
             SEXP x19 )  [inline]
```

Definition at line 287 of file lang.h.

References Rcpp_list19().

Here is the call graph for this function:



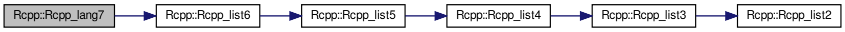
5.2.3.259 Rcpp_lang7()

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

Definition at line 79 of file lang.h.

References Rcpp_list6().

Here is the call graph for this function:



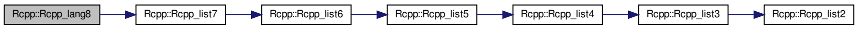
5.2.3.260 Rcpp_lang8()

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

Definition at line 95 of file lang.h.

References Rcpp_list7().

Here is the call graph for this function:



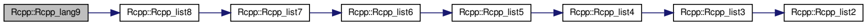
5.2.3.261 Rcpp_lang9()

```
SEXP Rcpp::Rcpp_lang9 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8 )  [inline]
```

Definition at line 111 of file lang.h.

References Rcpp_list8().

Here is the call graph for this function:



5.2.3.262 Rcpp_list10()

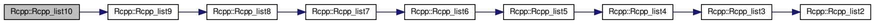
```
SEXP Rcpp::Rcpp_list10 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9 )  [inline]
```

Definition at line 119 of file lang.h.

References Rcpp_list9().

Referenced by Rcpp_lang11(), and Rcpp_list11().

Here is the call graph for this function:



5.2.3.263 Rcpp_list11()

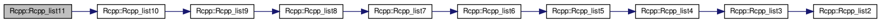
```
SEXP Rcpp::Rcpp_list11 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10 )  [inline]
```

Definition at line 135 of file lang.h.

References Rcpp_list10().

Referenced by Rcpp_lang12(), and Rcpp_list12().

Here is the call graph for this function:



5.2.3.264 Rcpp_list12()

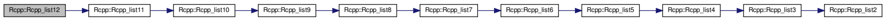
```
SEXP Rcpp::Rcpp_list12 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11 )  [inline]
```

Definition at line 151 of file lang.h.

References Rcpp_list11().

Referenced by Rcpp_lang13(), and Rcpp_list13().

Here is the call graph for this function:



5.2.3.265 Rcpp_list13()

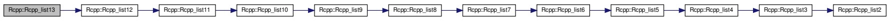
```
SEXP Rcpp::Rcpp_list13 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12 )  [inline]
```

Definition at line 167 of file lang.h.

References Rcpp_list12().

Referenced by Rcpp_lang14(), and Rcpp_list14().

Here is the call graph for this function:



5.2.3.266 Rcpp_list14()

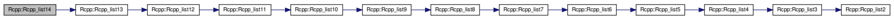
```
SEXP Rcpp::Rcpp_list14 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13 )  [inline]
```

Definition at line 183 of file lang.h.

References Rcpp_list13().

Referenced by Rcpp_lang15(), and Rcpp_list15().

Here is the call graph for this function:



5.2.3.267 Rcpp_list15()

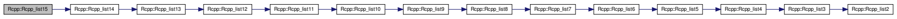
```
SEXP Rcpp::Rcpp_list15 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14 )  [inline]
```

Definition at line 199 of file lang.h.

References Rcpp_list14().

Referenced by Rcpp_lang16(), and Rcpp_list16().

Here is the call graph for this function:



5.2.3.268 Rcpp_list16()

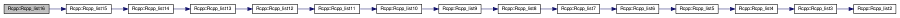
```
SEXP Rcpp::Rcpp_list16 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15 )  [inline]
```

Definition at line 215 of file lang.h.

References Rcpp_list15().

Referenced by Rcpp_lang17(), and Rcpp_list17().

Here is the call graph for this function:



5.2.3.269 Rcpp_list17()

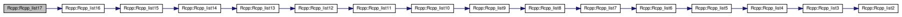
```
SEXP Rcpp::Rcpp_list17 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16 )  [inline]
```

Definition at line 231 of file lang.h.

References Rcpp_list16().

Referenced by Rcpp_lang18(), and Rcpp_list18().

Here is the call graph for this function:



5.2.3.270 Rcpp_list18()

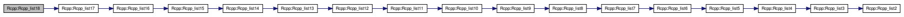
```
SEXP Rcpp::Rcpp_list18 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16,
             SEXP x17 )  [inline]
```

Definition at line 247 of file lang.h.

References Rcpp_list17().

Referenced by Rcpp_lang19(), and Rcpp_list19().

Here is the call graph for this function:



5.2.3.271 Rcpp_list19()

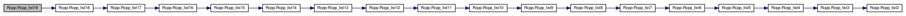
```
SEXP Rcpp::Rcpp_list19 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16,
             SEXP x17,
             SEXP x18 )  [inline]
```

Definition at line 263 of file lang.h.

References Rcpp_list18().

Referenced by Rcpp_lang20(), and Rcpp_list20().

Here is the call graph for this function:



5.2.3.272 Rcpp_list2()

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

Definition at line 36 of file lang.h.

References Rcpp_list1.

Referenced by Rcpp_list3().

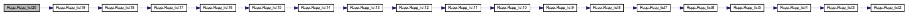
5.2.3.273 Rcpp_list20()

```
SEXP Rcpp::Rcpp_list20 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8,
             SEXP x9,
             SEXP x10,
             SEXP x11,
             SEXP x12,
             SEXP x13,
             SEXP x14,
             SEXP x15,
             SEXP x16,
             SEXP x17,
             SEXP x18,
             SEXP x19 )  [inline]
```

Definition at line 279 of file lang.h.

References Rcpp_list19().

Here is the call graph for this function:



5.2.3.274 Rcpp_list3()

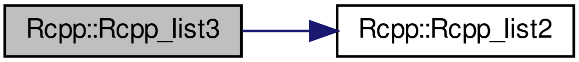
```
SEXP Rcpp::Rcpp_list3 (
             SEXP x0,
             SEXP x1,
             SEXP x2 )  [inline]
```

Definition at line 43 of file lang.h.

References Rcpp_list2().

Referenced by Rcpp_list4().

Here is the call graph for this function:



5.2.3.275 Rcpp_list4()

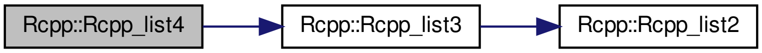
```
SEXP Rcpp::Rcpp_list4 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3 )  [inline]
```

Definition at line 50 of file lang.h.

References Rcpp_list3().

Referenced by Rcpp_list5().

Here is the call graph for this function:



5.2.3.276 Rcpp_list5()

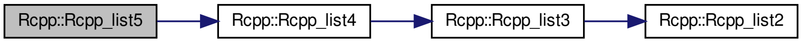
```
SEXP Rcpp::Rcpp_list5 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4 )  [inline]
```

Definition at line 57 of file lang.h.

References Rcpp_list4().

Referenced by Rcpp_list6().

Here is the call graph for this function:



5.2.3.277 Rcpp_list6()

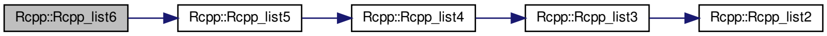
```
SEXP Rcpp::Rcpp_list6 (
            SEXP x0,
            SEXP x1,
            SEXP x2,
            SEXP x3,
            SEXP x4,
            SEXP x5 )  [inline]
```

Definition at line 64 of file lang.h.

References Rcpp_list5().

Referenced by Rcpp_lang7(), and Rcpp_list7().

Here is the call graph for this function:



5.2.3.278 Rcpp_list7()

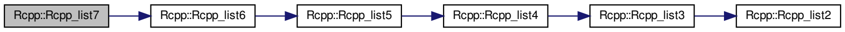
```
SEXP Rcpp::Rcpp_list7 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6 )  [inline]
```

Definition at line 71 of file lang.h.

References Rcpp_list6().

Referenced by Rcpp_lang8(), and Rcpp_list8().

Here is the call graph for this function:



5.2.3.279 Rcpp_list8()

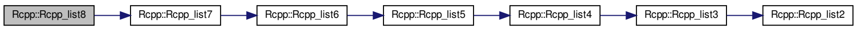
```
SEXP Rcpp::Rcpp_list8 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7 )  [inline]
```

Definition at line 87 of file lang.h.

References Rcpp_list7().

Referenced by Rcpp_lang9(), and Rcpp_list9().

Here is the call graph for this function:



5.2.3.280 Rcpp_list9()

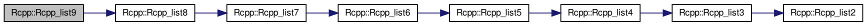
```
SEXP Rcpp::Rcpp_list9 (
             SEXP x0,
             SEXP x1,
             SEXP x2,
             SEXP x3,
             SEXP x4,
             SEXP x5,
             SEXP x6,
             SEXP x7,
             SEXP x8 )  [inline]
```

Definition at line 103 of file lang.h.

References Rcpp_list8().

Referenced by Rcpp_lang10(), and Rcpp_list10().

Here is the call graph for this function:



5.2.3.281 Rcpp_namespace()

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

Returns

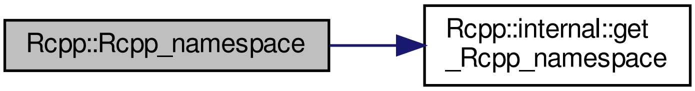
the Rcpp namespace

Definition at line 360 of file Environment.h.

References Rcpp::internal::get_Rcpp_namespace().

Referenced by Rcpp::internal::make_new_object(), and RCPP_API_CLASS().

Here is the call graph for this function:



5.2.3.282 Rcpp_precious_init()

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

Definition at line 142 of file routines.h.

References GET_CALLABLE.

Referenced by R_init_Rcpp(), and registerFunctions().

5.2.3.283 Rcpp_precious_preserve()

```
SEXP Rcpp::Rcpp_precious_preserve (
             SEXP object )  [inline]
```

Definition at line 152 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp_PreciousPreserve(), and registerFunctions().

5.2.3.284 Rcpp_precious_remove()

```
void Rcpp::Rcpp_precious_remove (
             SEXP token )  [inline]
```

Definition at line 157 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp_PreciousRelease(), and registerFunctions().

5.2.3.285 Rcpp_precious_teardown()

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

Definition at line 147 of file routines.h.

References GET_CALLABLE.

Referenced by R_unload_Rcpp(), and registerFunctions().

5.2.3.286 Rcpp_PreciousPreserve()

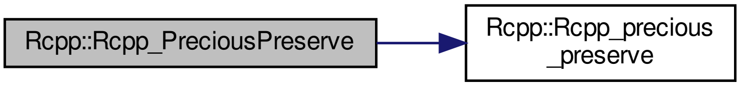
```
SEXP Rcpp::Rcpp_PreciousPreserve (
             SEXP object )  [inline]
```

Definition at line 116 of file RcppCommon.h.

References Rcpp_precious_preserve().

Referenced by Rcpp::String::append_wide_string(), Rcpp::String::assign_wide_string(), Rcpp::traits::named_object< SEXP >::named_object(), Rcpp::String::operator+=(), Rcpp::String::operator=(), Rcpp::PreserveStorage< CLASS >↩ ::set__(), Rcpp::String::set_encoding(), Rcpp::String::set_na(), Rcpp::String::setData(), and Rcpp::String::String().

Here is the call graph for this function:



5.2.3.287 Rcpp_PreciousRelease()

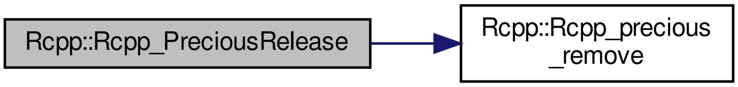
```
void Rcpp::Rcpp_PreciousRelease (
             SEXP token )  [inline]
```

Definition at line 120 of file RcppCommon.h.

References Rcpp_precious_remove().

Referenced by Rcpp::String::append_wide_string(), Rcpp::String::assign_wide_string(), Rcpp::PreserveStorage< CLASS >::invalidate__(), Rcpp::String::operator+=(), Rcpp::String::operator=(), Rcpp::PreserveStorage< CLASS >::set__(), Rcpp::String::set_encoding(), Rcpp::String::set_na(), Rcpp::traits::named_object< SEXP >::∼named_↩ object(), Rcpp::PreserveStorage< CLASS >::∼PreserveStorage(), and Rcpp::String::∼String().

Here is the call graph for this function:



5.2.3.288 Rcpp_PreserveObject()

```
SEXP Rcpp::Rcpp_PreserveObject (
             SEXP x )  [inline]
```

Definition at line 98 of file RcppCommon.h.

Referenced by Rcpp_ReplaceObject().

5.2.3.289 Rcpp_protect()

```
SEXP Rcpp::Rcpp_protect (
             SEXP x )  [inline]
```

Definition at line 23 of file Shield.h.

Referenced by Rcpp::Shelter< T >::operator()().

5.2.3.290 Rcpp_ReleaseObject()

```
void Rcpp::Rcpp_ReleaseObject (
             SEXP x )  [inline]
```

Definition at line 102 of file RcppCommon.h.

Referenced by Rcpp_ReplaceObject().

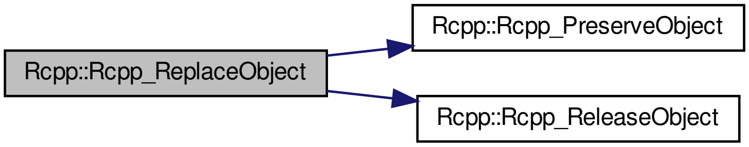
5.2.3.291 Rcpp_ReplaceObject()

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

Definition at line 105 of file RcppCommon.h.

References Rcpp_PreserveObject(), and Rcpp_ReleaseObject().

Here is the call graph for this function:



5.2.3.292 RCPP_SIMPLE_EXCEPTION_CLASS()

```
Rcpp::RCPP_SIMPLE_EXCEPTION_CLASS (
            not_a_matrix ,
            "Not a matrix."  )
```

5.2.3.293 Rcpp_unprotect()

```
void Rcpp::Rcpp_unprotect (
            int i )  [inline]
```

Definition at line 28 of file Shield.h.

Referenced by Rcpp::Shelter< T >::∼Shelter(), and Rcpp::Shield< T >::∼Shield().

5.2.3.294 remove()

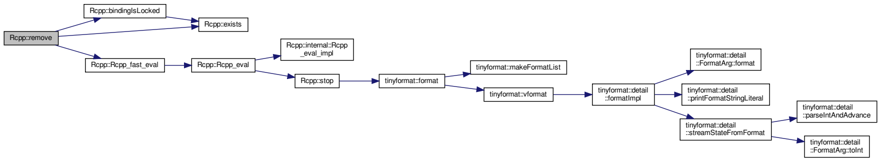
```
bool Rcpp::remove (
            const std::string & name )
```

remove an object from this environment

Definition at line 240 of file Environment.h.

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

Here is the call graph for this function:


Rcpp::remove
Rcpp::bindingIsLocked
Rcpp::exists
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

5.2.3.295 rep() [1/6]

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

Definition at line 85 of file rep.h.

5.2.3.296 rep() [2/6]

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

Definition at line 73 of file rep.h.

5.2.3.297 rep() [3/6]

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

Definition at line 76 of file rep.h.

5.2.3.298 rep() [4/6]

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

Definition at line 79 of file rep.h.

5.2.3.299 rep() [5/6]

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

Definition at line 82 of file rep.h.

5.2.3.300 rep() [6/6]

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

Definition at line 69 of file rep.h.

5.2.3.301 rep_each()

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

Definition at line 50 of file rep_each.h.

5.2.3.302 rep_len()

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

Definition at line 50 of file rep_len.h.

5.2.3.303 rev()

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

Definition at line 50 of file rev.h.

5.2.3.304 rexp() [1/2]

```
NumericVector Rcpp::rexp (
             int n )  [inline]
```

Definition at line 147 of file random.h.

5.2.3.305 rexp() [2/2]

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

Definition at line 137 of file random.h.

5.2.3.306 rf()

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

Definition at line 151 of file random.h.

5.2.3.307 rgamma() [1/2]

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

Definition at line 174 of file random.h.

5.2.3.308 rgamma() [2/2]

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

Definition at line 165 of file random.h.

5.2.3.309 rgeom()

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

Definition at line 183 of file random.h.

5.2.3.310 rhyper()

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

Definition at line 189 of file random.h.

5.2.3.311 rlnorm() [1/3]

```
NumericVector Rcpp::rlnorm (
             int n )  [inline]
```

Definition at line 215 of file random.h.

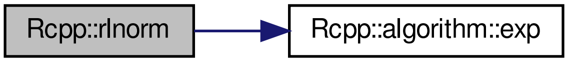
5.2.3.312 rlnorm() [2/3]

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

Definition at line 204 of file random.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:



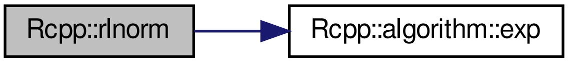
5.2.3.313 rlnorm() [3/3]

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

Definition at line 193 of file random.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:



5.2.3.314 rlogis() [1/3]

```
NumericVector Rcpp::rlogis (
            int n )  [inline]
```

Definition at line 239 of file random.h.

5.2.3.315 rlogis() [2/3]

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

Definition at line 229 of file random.h.

5.2.3.316 rlogis() [3/3]

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

Definition at line 219 of file random.h.

5.2.3.317 rnbinom()

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

Definition at line 243 of file random.h.

5.2.3.318 rnbinom_mu()

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

Definition at line 251 of file random.h.

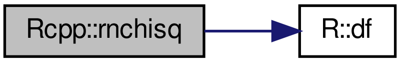
5.2.3.319 rnchisq() [1/2]

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

Definition at line 268 of file random.h.

References R::df().

Here is the call graph for this function:



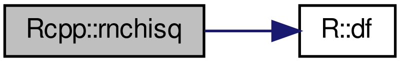
5.2.3.320 rnchisq() [2/2]

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

Definition at line 258 of file random.h.

References R::df().

Here is the call graph for this function:



5.2.3.321 rnorm() [1/3]

```
NumericVector Rcpp::rnorm (
             int n )  [inline]
```

Definition at line 96 of file random.h.

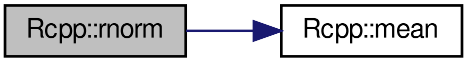
5.2.3.322 rnorm() [2/3]

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

Definition at line 80 of file random.h.

References mean().

Here is the call graph for this function:



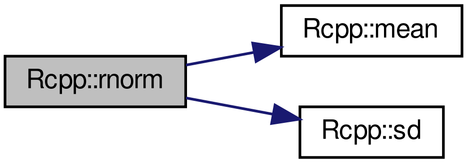
5.2.3.323 rnorm() [3/3]

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

Definition at line 58 of file random.h.

References mean(), and sd().

Here is the call graph for this function:



5.2.3.324 row()

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

Definition at line 55 of file row.h.

Referenced by operator<<().

5.2.3.325 rowMeans()

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

Definition at line 942 of file rowSums.h.

5.2.3.326 rownames()

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

Definition at line 209 of file Matrix.h.

5.2.3.327 rowSums()

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

Definition at line 924 of file rowSums.h.

5.2.3.328 rpois()

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

Definition at line 274 of file random.h.

5.2.3.329 rsignrank()

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

Definition at line 278 of file random.h.

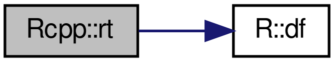
5.2.3.330 rt()

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

Definition at line 282 of file random.h.

References R::df().

Here is the call graph for this function:



5.2.3.331 runif() [1/3]

```
NumericVector Rcpp::runif (
            int n )  [inline]
```

Definition at line 307 of file random.h.

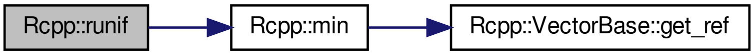
5.2.3.332 runif() [2/3]

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

Definition at line 301 of file random.h.

References min().

Here is the call graph for this function:



5.2.3.333 runif() [3/3]

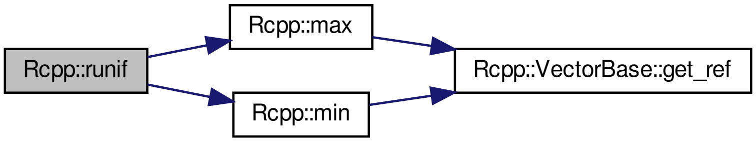
```
NumericVector Rcpp::runif (
             int n,
             double min,
             double max )  [inline]
```

Definition at line 295 of file random.h.

References max(), and min().

Referenced by piSugar().

Here is the call graph for this function:



5.2.3.334 rweibull() [1/2]

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

Definition at line 320 of file random.h.

5.2.3.335 rweibull() [2/2]

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

Definition at line 311 of file random.h.

5.2.3.336 rwilcox()

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

Definition at line 327 of file random.h.

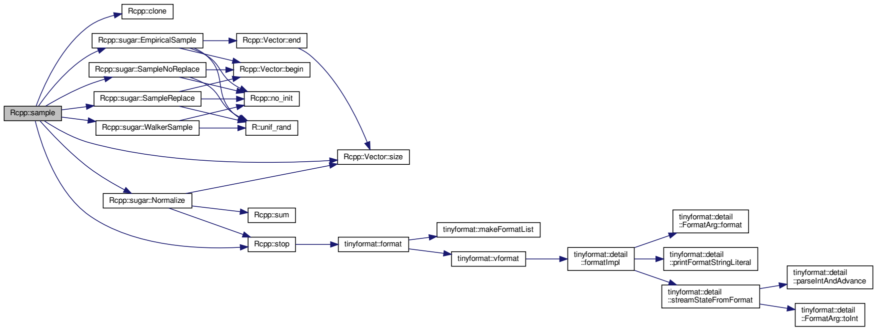
5.2.3.337 sample() [1/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sample (
              const Vector< RTYPE > & x,
              int size,
              bool replace = false,
              sugar::probs_t probs = R_NilValue )  [inline]
```

Definition at line 437 of file sample.h.

References clone(), Rcpp::sugar::EmpiricalSample(), Rcpp::sugar::Normalize(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), and Rcpp::sugar::WalkerSample().

Here is the call graph for this function:


Rcpp::sample
Rcpp::clone
Rcpp::sugar::EmpiricalSample
Rcpp::sugar::SampleNoReplace
Rcpp::sugar::SampleReplace
Rcpp::sugar::WalkerSample
Rcpp::Vector::end
Rcpp::Vector::begin
Rcpp::no_init
R::unif_rand
Rcpp::Vector::size
Rcpp::sugar::Normalize
Rcpp::sum
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

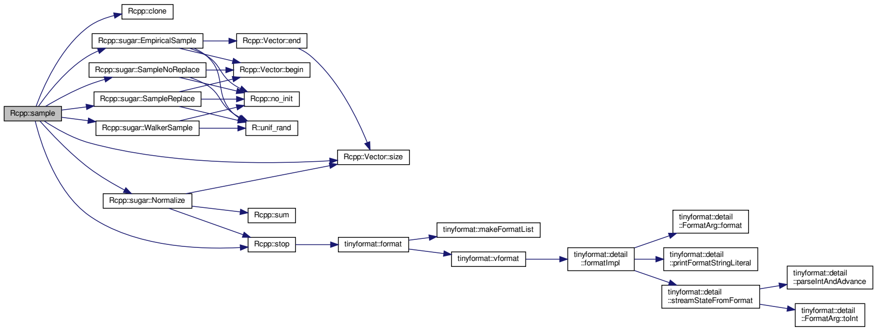
5.2.3.338 sample() [2/2]

```
Vector<INTSXP> Rcpp::sample (
             int n,
             int size,
             bool replace = false,
             sugar::probs_t probs = R_NilValue,
             bool one_based = true )  [inline]
```

Definition at line 401 of file sample.h.

References clone(), Rcpp::sugar::EmpiricalSample(), Rcpp::sugar::Normalize(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), and Rcpp::sugar::Walker↩ Sample().

Here is the call graph for this function:


Rcpp::sample
Rcpp::clone
Rcpp::sugar::EmpiricalSample
Rcpp::sugar::SampleNoReplace
Rcpp::sugar::SampleReplace
Rcpp::sugar::WalkerSample
Rcpp::Vector::end
Rcpp::Vector::begin
Rcpp::no_init
R::unif_rand
Rcpp::Vector::size
Rcpp::sugar::Normalize
Rcpp::sum
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

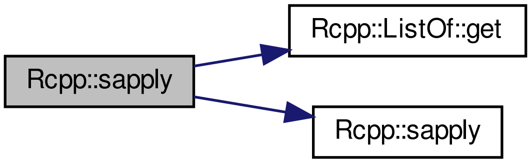
5.2.3.339 sapply() [1/2]

```
template<typename T , typename Function >
T Rcpp::sapply (
             const ListOf< T > & t,
             Function fun )
```

Definition at line 134 of file ListOf.h.

References Rcpp::ListOf< T >::get(), and sapply().

Here is the call graph for this function:



5.2.3.340 sapply() [2/2]

```
template<int RTYPE, bool NA, typename T , typename Function >
sugar::Sapply< RTYPE,NA,T,Function, traits::same_type< typename ::Rcpp::sugar::sapply_application_result_of<Func
T>::type , typename Rcpp::traits::storage_type< traits::r_sexptype_traits< typename ::Rcpp::sugar::sapply_applic
T>::type >::rtype >::type >::value> Rcpp::sapply (
             const Rcpp::VectorBase< RTYPE, NA, T > & t,
             Function fun )  [inline]
```

Definition at line 126 of file sapply.h.

References NA.

Referenced by sapply().

5.2.3.341 sd()

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

Definition at line 46 of file sd.h.

Referenced by rnorm().

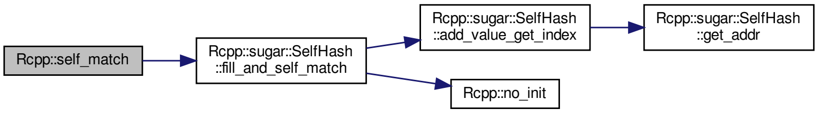
5.2.3.342 self_match()

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

Definition at line 69 of file self_match.h.

References Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match().

Here is the call graph for this function:


Rcpp::self_match
Rcpp::sugar::SelfHash ::fill_and_self_match
Rcpp::sugar::SelfHash ::add_value_get_index
Rcpp::sugar::SelfHash ::get_addr
Rcpp::no_init

5.2.3.343 seq()

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

Definition at line 52 of file seq_along.h.

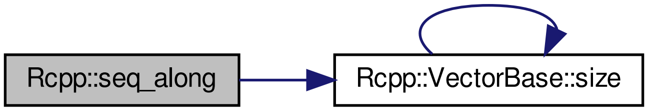
5.2.3.344 seq_along()

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

Definition at line 44 of file seq_along.h.

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

Here is the call graph for this function:



5.2.3.345 seq_len()

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

Definition at line 48 of file seq_along.h.

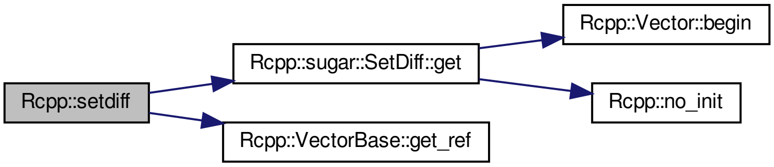
5.2.3.346 setdiff()

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

Definition at line 161 of file setdiff.h.

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

Here is the call graph for this function:



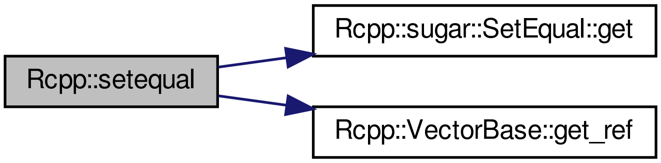
5.2.3.347 setequal()

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

Definition at line 166 of file setdiff.h.

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

Here is the call graph for this function:



5.2.3.348 setFunction()

```
void Rcpp::setFunction (
             const Function & function )
```

sets the function

Definition at line 126 of file Language.h.

5.2.3.349 setSymbol() [1/2]

```
void Rcpp::setSymbol (
             const std::string & symbol )
```

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

Parameters

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 110 of file Language.h.

5.2.3.350 setSymbol() [2/2]

```
void Rcpp::setSymbol (
             const Symbol & symbol )
```

sets the symbol of the call

Definition at line 117 of file Language.h.

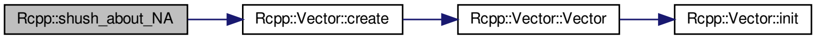
5.2.3.351 shush_about_NA()

```
LogicalVector Rcpp::shush_about_NA ( )  [inline]
```

Definition at line 54 of file Na_Proxy.h.

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

Here is the call graph for this function:



5.2.3.352 sign() [1/2]

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

Definition at line 70 of file sign.h.

Referenced by Rcpp::Module::get_function().

5.2.3.353 sign() [2/2]

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

Definition at line 75 of file sign.h.

5.2.3.354 sort_unique()

```
template<int RTYPE, bool NA, typename T >
Vector<RTYPE> Rcpp::sort_unique (
            const VectorBase< RTYPE, NA, T > & t,
            bool decreasing = false )  [inline]
```

Definition at line 72 of file unique.h.

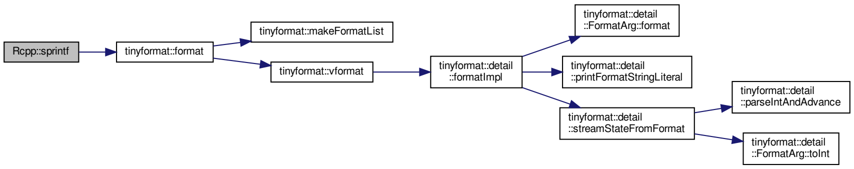
5.2.3.355 sprintf()

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

Definition at line 28 of file sprintf.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::sprintf
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.2.3.356 standard_delete_finalizer()

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

Definition at line 30 of file XPtr.h.

5.2.3.357 stop() [1/12]

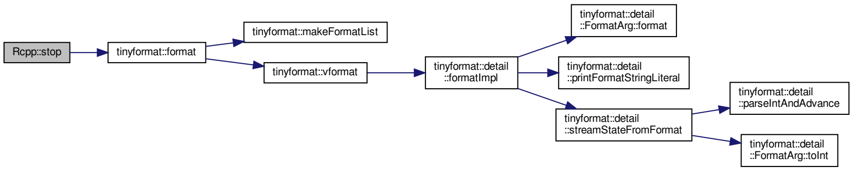
```
template<typename... Args>
void NORET Rcpp::stop (
            const char * fmt,
            Args &&... args ) [inline]
```

Definition at line 51 of file exceptions.h.

References tinyformat::format().

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

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

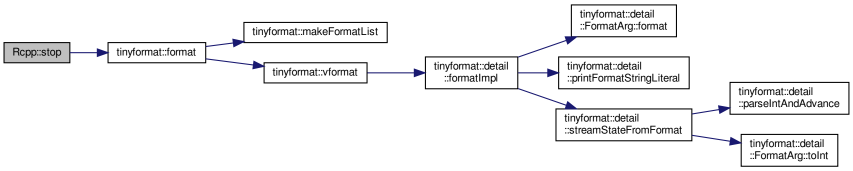
5.2.3.358 stop() [2/12]

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

Definition at line 126 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

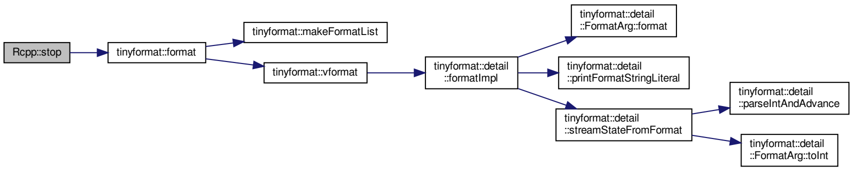
5.2.3.359 stop() [3/12]

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

Definition at line 131 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

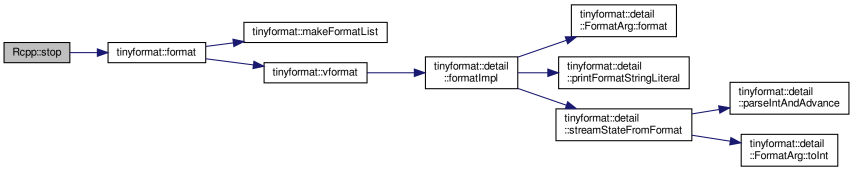
5.2.3.360 stop() [4/12]

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

Definition at line 136 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

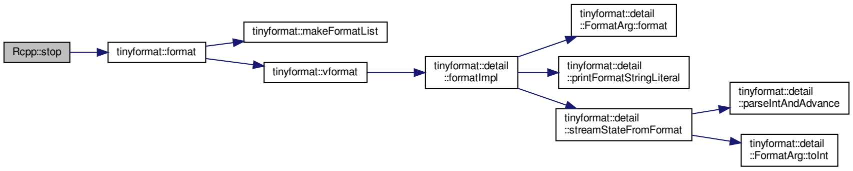
5.2.3.361 stop() [5/12]

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

Definition at line 141 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

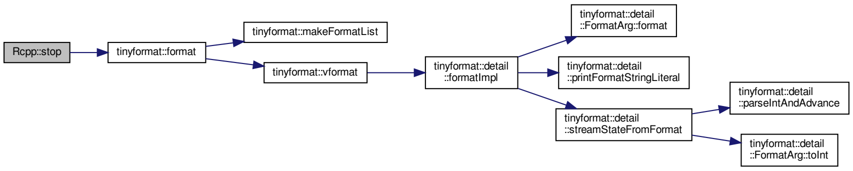
5.2.3.362 stop() [6/12]

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

Definition at line 146 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

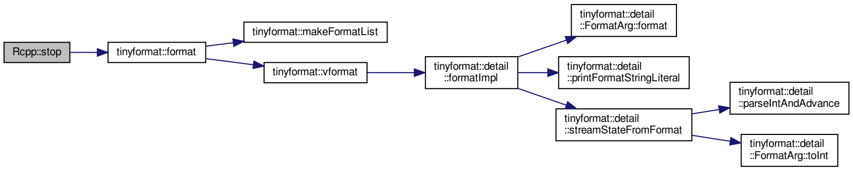
5.2.3.363 stop() [7/12]

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

Definition at line 151 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

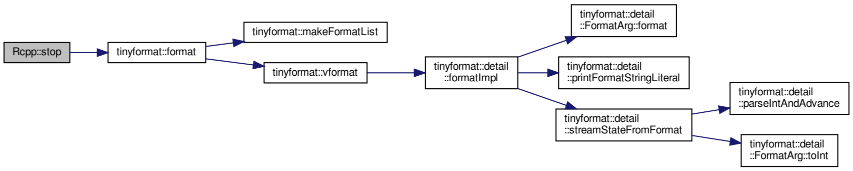
5.2.3.364 stop() [8/12]

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

Definition at line 156 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

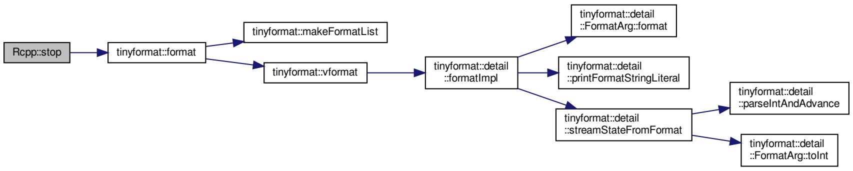
5.2.3.365 stop() [9/12]

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

Definition at line 161 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

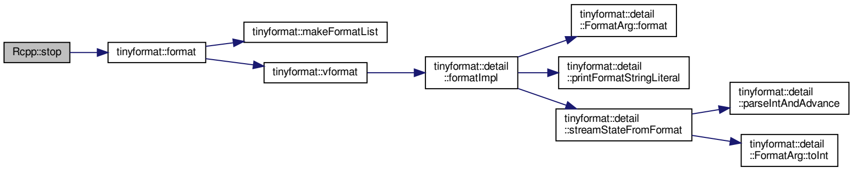
5.2.3.366 stop() [10/12]

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

Definition at line 166 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

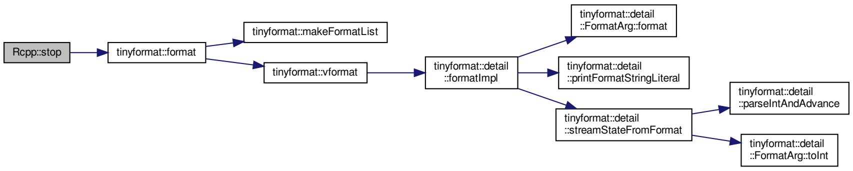
5.2.3.367 stop() [11/12]

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

Definition at line 171 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

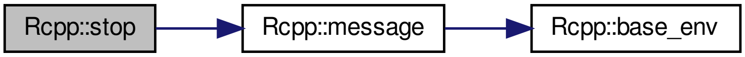
5.2.3.368 stop() [12/12]

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

Definition at line 117 of file exceptions.h.

References message().

Here is the call graph for this function:



5.2.3.369 StretchyList_Impl() [1/2]

```
Rcpp::StretchyList_Impl ( )
```

Definition at line 42 of file StretchyList.h.

5.2.3.370 StretchyList_Impl() [2/2]

```
Rcpp::StretchyList_Impl (
             SEXP x )
```

Definition at line 47 of file StretchyList.h.

5.2.3.371 sum() [1/3]

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

Definition at line 98 of file sum.h.

Referenced by Rcpp::sugar::na_omit_impl(), Rcpp::sugar::Normalize(), and piSugar().

5.2.3.372 sum() [2/3]

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

Definition at line 108 of file sum.h.

5.2.3.373 sum() [3/3]

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

Definition at line 103 of file sum.h.

5.2.3.374 table()

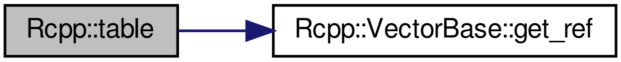
```
template<int RTYPE, bool NA, typename T >
IntegerVector Rcpp::table (
             const VectorBase< RTYPE, NA, T > & x )  [inline]
```

Definition at line 126 of file table.h.

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

Referenced by in(), match(), Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch(), and Rcpp::sugar::Table< RTYPE, TABLE_T >::Table().

Here is the call graph for this function:



5.2.3.375 tail()

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

Definition at line 56 of file tail.h.

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

5.2.3.376 timesub()

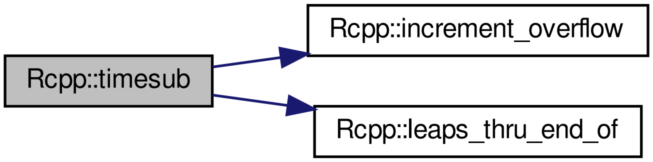
```
static struct tm * Rcpp::timesub (
             const time_t * timep,
             int_fast32_t offset,
             const struct state * sp,
             struct tm * tmp )  [static]
```

Definition at line 1227 of file date.cpp.

References DAYSPERLYEAR, 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, mon_lengths, SECSPERDAY, SECSPERHOUR, SECSPERMIN, TM_YEAR_BASE, TYPE_SIGNED, and year↩
_lengths.

Referenced by gmtsub().

Here is the call graph for this function:



5.2.3.377 toString()

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

Definition at line 64 of file exceptions.h.

5.2.3.378 tranpose_impl()

```
template<int RTYPE, template< class > class StoragePolicy>
Matrix<RTYPE, StoragePolicy> Rcpp::tranpose_impl (
             const Matrix< RTYPE, StoragePolicy > & x )
```

Definition at line 406 of file Matrix.h.

5.2.3.379 transpose() [1/3]

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

Definition at line 440 of file Matrix.h.

5.2.3.380 transpose() [2/3]

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

Definition at line 435 of file Matrix.h.

5.2.3.381 transpose() [3/3]

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

Definition at line 445 of file Matrix.h.

5.2.3.382 transtime()

```
static int_fast32_t Rcpp::transtime (
             int year,
             const struct rule * rulep,
             int_fast32_t offset ) [static]
```

Definition at line 879 of file date.cpp.

References DAY_OF_YEAR, DAYSPERWEEK, INITIALIZE, isleap, JULIAN_DAY, mon_lengths, MONTH_NTH_DAY_↩OF_WEEK, Rcpp::rule::r_day, Rcpp::rule::r_mon, Rcpp::rule::r_time, Rcpp::rule::r_type, Rcpp::rule::r_week, and SECSPERDAY.

Referenced by tzparse().

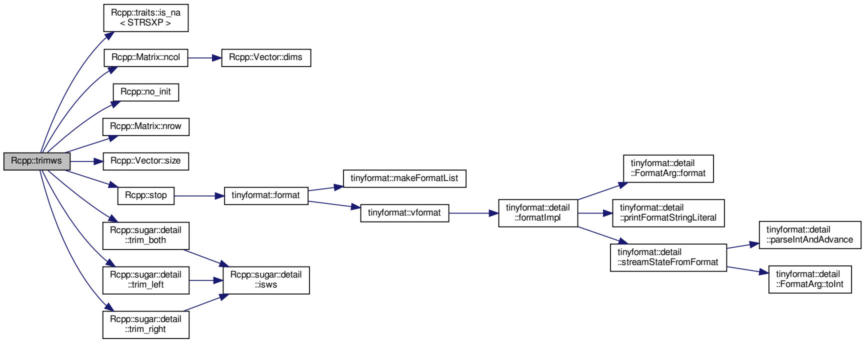
5.2.3.383 trimws() [1/3]

```
Matrix<STRSXP> Rcpp::trimws (
              const Matrix< STRSXP > & x,
              const char * which = "both" )  [inline]
```

Definition at line 135 of file trimws.h.

References Rcpp::traits::is_na< STRSXP >(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), no_init(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), Rcpp::sugar::detail::trim_both(), Rcpp::sugar::detail::trim_left(), and Rcpp::sugar::detail::trim_right().

Here is the call graph for this function:


Rcpp::traits::is_na < STRSXP >
Rcpp::Matrix::ncol
Rcpp::Vector::dims
Rcpp::no_init
Rcpp::Matrix::nrow
Rcpp::trimws
Rcpp::Vector::size
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt
Rcpp::sugar::detail ::trim_both
Rcpp::sugar::detail ::trim_left
Rcpp::sugar::detail ::isws
Rcpp::sugar::detail ::trim_right

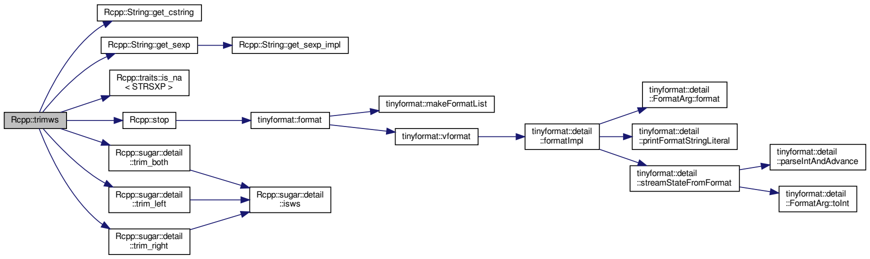
5.2.3.384 trimws() [2/3]

```
String Rcpp::trimws (
              const String & str,
              const char * which = "both" )  [inline]
```

Definition at line 181 of file trimws.h.

References Rcpp::String::get_cstring(), Rcpp::String::get_sexp(), Rcpp::traits::is_na< STRSXP >(), stop(), Rcpp::sugar::detail::trim_both(), Rcpp::sugar::detail::trim_left(), and Rcpp::sugar::detail::trim_right().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::get_sexp
Rcpp::String::get_sexp_impl
Rcpp::traits::is_na < STRSXP >
Rcpp::trimws
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt
Rcpp::sugar::detail ::trim_both
Rcpp::sugar::detail ::trim_left
Rcpp::sugar::detail ::isws
Rcpp::sugar::detail ::trim_right

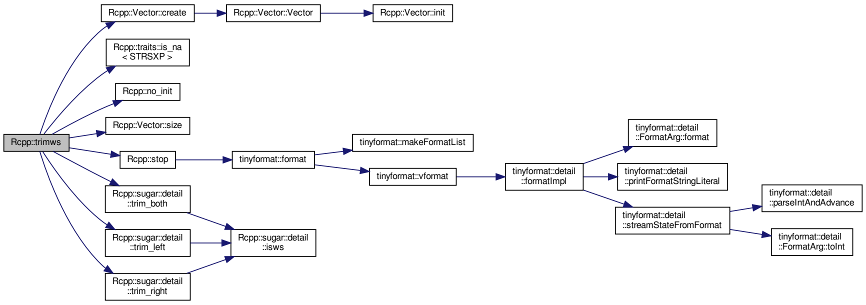
5.2.3.385 trimws() [3/3]

```
Vector<STRSXP> Rcpp::trimws (
             const Vector< STRSXP > & x,
             const char * which = "both" )  [inline]
```

Definition at line 90 of file trimws.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::traits::is_na< STRSXP >(), no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), Rcpp::sugar::detail::trim_both(), Rcpp::sugar::detail::trim_left(), and Rcpp::sugar::detail::trim_right().

Here is the call graph for this function:


Rcpp::trimws
Rcpp::Vector::create
Rcpp::Vector::Vector
Rcpp::Vector::init
Rcpp::traits::is_na < STRSXP >
Rcpp::no_init
Rcpp::Vector::size
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt
Rcpp::sugar::detail ::trim_both
Rcpp::sugar::detail ::trim_left
Rcpp::sugar::detail ::trim_right
Rcpp::sugar::detail ::isws

5.2.3.386 type2name()

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

Definition at line 84 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.2.3.387 typesequiv()

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

Definition at line 1202 of file date.cpp.

References Rcpp::state::chars, Rcpp::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().

5.2.3.388 tzload()

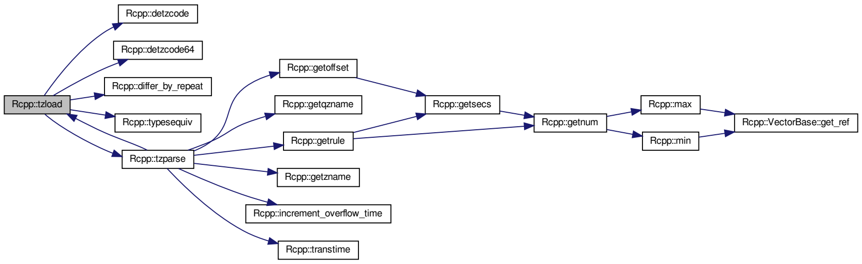
```
static int Rcpp::tzload (
             const char * name,
             struct state *const sp,
             const int doextend )  [static]
```

Definition at line 632 of file date.cpp.

References Rcpp::state::ats, Rcpp::state::charcnt, Rcpp::state::chars, detzcode(), detzcode64(), differ_by_repeat(), Rcpp::state::goahead, Rcpp::state::goback, Rcpp::state::leapcnt, Rcpp::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:



5.2.3.389 tzparse()

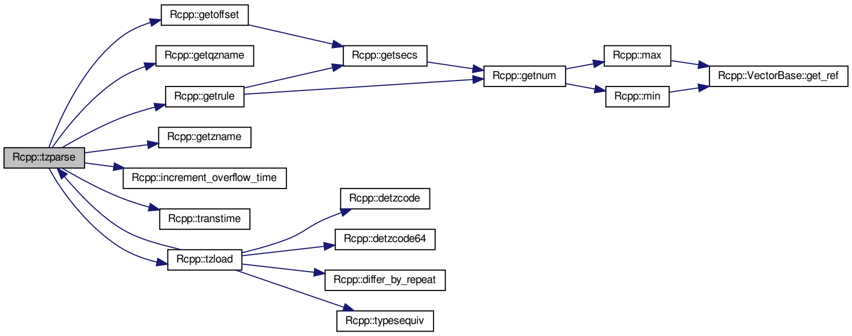
```
static int Rcpp::tzparse (
             const char * name,
             struct state * sp,
             int lastditch )  [static]
```

Definition at line 966 of file date.cpp.

References Rcpp::state::ats, Rcpp::state::charcnt, Rcpp::state::chars, EPOCH_YEAR, getoffset(), getqzname(), getrule(), getzname(), increment_overflow_time(), 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, tzload(), year_lengths, and YEARSPERREPEAT.

Referenced by gmtload(), and tzload().

Here is the call graph for this function:



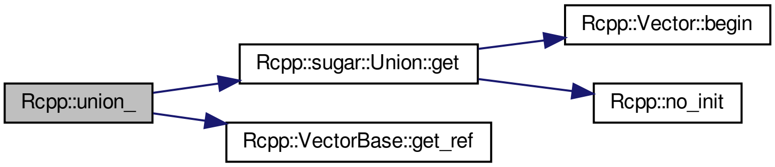
5.2.3.390 union_()

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

Definition at line 177 of file setdiff.h.

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

Here is the call graph for this function:



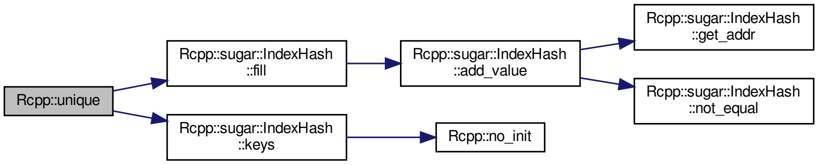
5.2.3.391 unique()

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

Definition at line 65 of file unique.h.

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

Here is the call graph for this function:


Rcpp::unique
Rcpp::sugar::IndexHash ::fill
Rcpp::sugar::IndexHash ::keys
Rcpp::sugar::IndexHash ::add_value
Rcpp::no_init
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

5.2.3.392 unlockBinding()

```
void Rcpp::unlockBinding (
             const std::string & name )
```

unlocks the given binding see ?bindingIsLocked

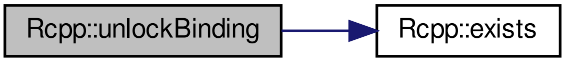
Exceptions

no_such_binding	if there is no such binding in this environment

Definition at line 286 of file Environment.h.

References exists().

Here is the call graph for this function:



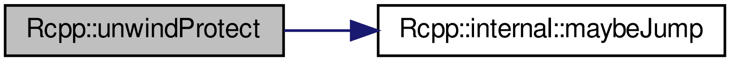
5.2.3.393 unwindProtect()

```
SEXP Rcpp::unwindProtect (
            SEXP(*)(void *data) callback,
            void * data )  [inline]
```

Definition at line 60 of file unwindProtect.h.

References Rcpp::internal::UnwindData::jmpbuf, and Rcpp::internal::maybeJump().

Here is the call graph for this function:



5.2.3.394 update()

```
void Rcpp::update (
            SEXP x )
```

Definition at line 400 of file Environment.h.

Referenced by RCPP_API_CLASS().

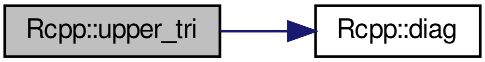
5.2.3.395 upper_tri()

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

Definition at line 63 of file upper_tri.h.

References diag().

Here is the call graph for this function:



5.2.3.396 var() [1/4]

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

Definition at line 89 of file var.h.

5.2.3.397 var() [2/4]

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

Definition at line 79 of file var.h.

5.2.3.398 var() [3/4]

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

Definition at line 84 of file var.h.

5.2.3.399 var() [4/4]

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

Definition at line 74 of file var.h.

Referenced by Rcpp::sugar::Sd< RTYPE, NA, T >::get().

5.2.3.400 warning() [1/12]

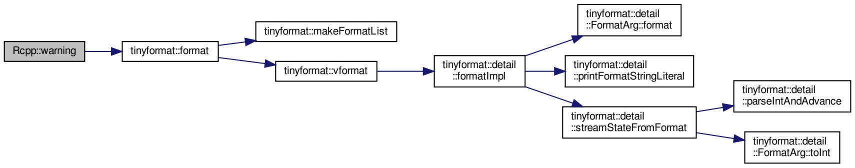
```
template<typename... Args>
void Rcpp::warning (
             const char * fmt,
             Args &&... args )  [inline]
```

Definition at line 46 of file exceptions.h.

References tinyformat::format().

Referenced by r_cast(), Rcpp::DataFrame_Impl< StoragePolicy >::set_type_after_push(), and Rcpp::attributes::showWarning().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

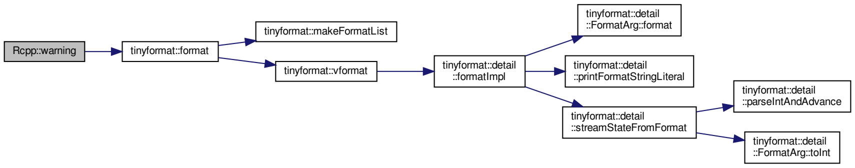
5.2.3.401 warning() [2/12]

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

Definition at line 72 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

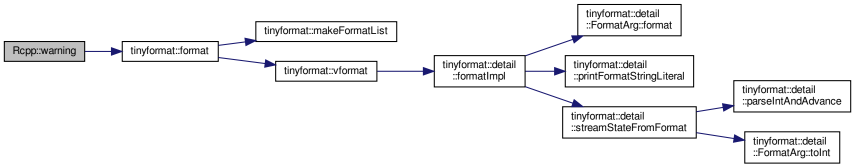
5.2.3.402 warning() [3/12]

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

Definition at line 77 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

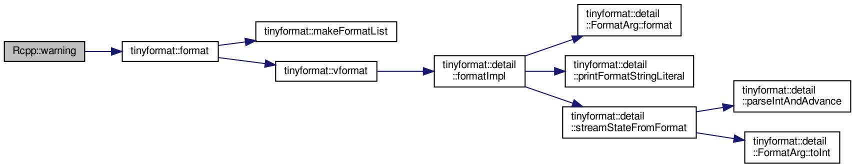
5.2.3.403 warning() [4/12]

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

Definition at line 82 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

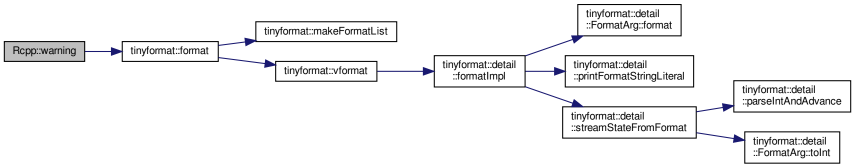
5.2.3.404 warning() [5/12]

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

Definition at line 87 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

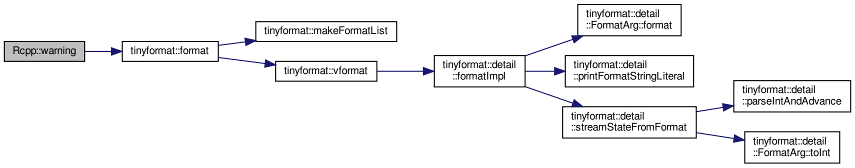
5.2.3.405 warning() [6/12]

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

Definition at line 92 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

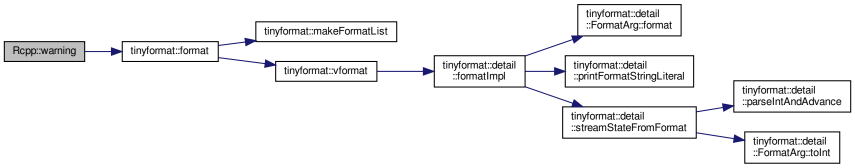
5.2.3.406 warning() [7/12]

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

Definition at line 97 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

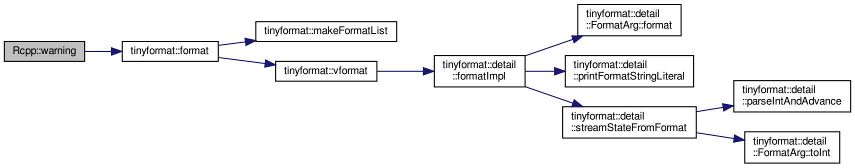
5.2.3.407 warning() [8/12]

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

Definition at line 102 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

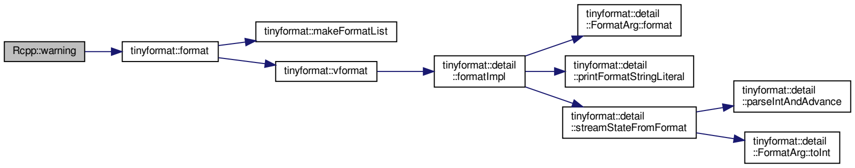
5.2.3.408 warning() [9/12]

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

Definition at line 107 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

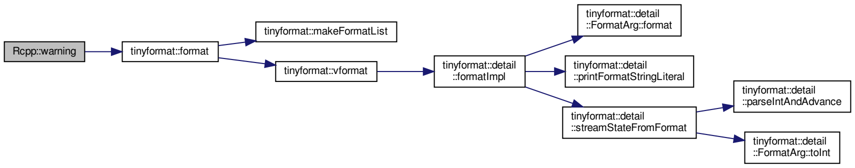
5.2.3.409 warning() [10/12]

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

Definition at line 112 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

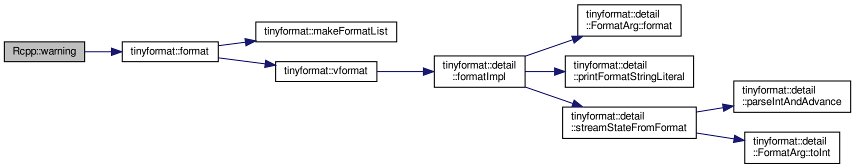
5.2.3.410 warning() [11/12]

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

Definition at line 117 of file exceptions.h.

References tinyformat::format().

Here is the call graph for this function:


Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

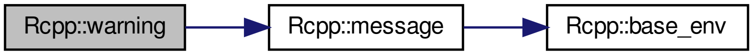
5.2.3.411 warning() [12/12]

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

Definition at line 113 of file exceptions.h.

References message().

Here is the call graph for this function:



5.2.3.412 warningcall()

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

Definition at line 29 of file print.h.

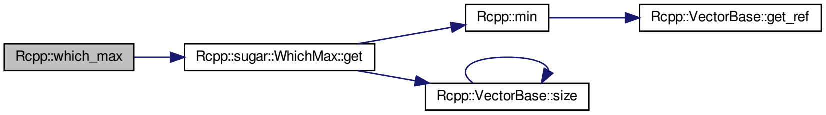
5.2.3.413 which_max()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::which_max (
             const VectorBase< RTYPE, NA, T > & t )
```

Definition at line 90 of file which_max.h.

References Rcpp::sugar::WhichMax< RTYPE, NA, T >::get().

Here is the call graph for this function:



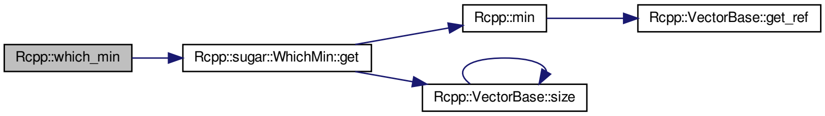
5.2.3.414 which_min()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::which_min (
             const VectorBase< RTYPE, NA, T > & t )
```

Definition at line 90 of file which_min.h.

References Rcpp::sugar::WhichMin< RTYPE, NA, T >::get().

Here is the call graph for this function:



5.2.3.415 wrap()

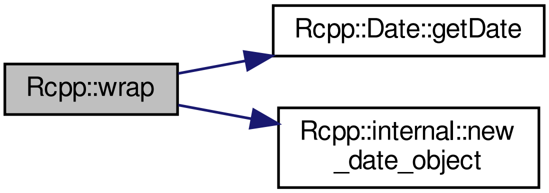
```
template<>
SEXP Rcpp::wrap (
            const Date & date )  [inline]
```

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< RTYPE, StoragePolicy >↩ ::assign(), Rcpp::Vector< RTYPE, StoragePolicy >::assign_object(), Rcpp::Vector< RTYPE, StoragePolicy >::assign↩ _sugar_expression(), Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl(), CppProperty_Getter_Setter< PROP >::get(), CppProperty_Getter< PROP >::get(), CppProperty_GetMethod< Class, PROP >::get(), Cpp↩ Property_GetConstMethod< Class, PROP >::get(), CppProperty_GetPointerMethod< Class, PROP >::get(), Cpp↩ Property_GetMethod_SetMethod< Class, PROP >::get(), CppProperty_GetConstMethod_SetMethod< Class, PROP >::get(), CppProperty_GetMethod_SetPointer< Class, PROP >::get(), CppProperty_GetConstMethod_SetPointer< Class, PROP >::get(), CppProperty_GetPointer_SetMethod< Class, PROP >::get(), CppProperty_GetPointer_↩ SetPointer< Class, PROP >::get(), Rcpp::internal::generic_element_converter< RTYPE >::get(), Rcpp::sugar↩ ::IndexHash< RTYPE >::get_profile_data(), Rcpp::internal::grow__dispatch(), Rcpp::Dimension::operator SEXP(), Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP(), Rcpp::ListOf< T >::operator SEXP(), Rcpp::internal::simple↩ _name_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_↩ RTYPE, RHS_NA, RHS_T >::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=(), Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=(), Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↩ ::operator=(), Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator=(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator=(), Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=(), Rcpp::FieldProxy↩ Policy< CLASS >::FieldProxy::operator=(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::ChildVector< T >::operator=(), Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator=(), Rcpp::Armor< T >::operator=(), Rcpp↩ ::sugar::Lapply< RTYPE, NA, T, Function >::operator[ ](), PiLeibniz(), Rcpp::internal::range_wrap_dispatch___impl_↩ _pair(), and RCPP_API_CLASS().

Here is the call graph for this function:



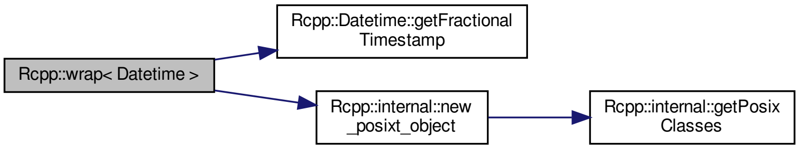
5.2.3.416 wrap< Datetime >()

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

Definition at line 44 of file Datetime.h.

References Rcpp::Datetime::getFractionalTimestamp(), and Rcpp::internal::new_posixt_object().

Here is the call graph for this function:


Rcpp::wrap< Datetime >
Rcpp::Datetime::getFractional Timestamp
Rcpp::internal::new _posixt_object
Rcpp::internal::getPosix Classes

5.2.3.417 wrap< Rcpp::Date >()

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

5.2.3.418 wrap< Rcpp::Datetime >()

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

5.2.3.419 wrap< Rcpp::String >()

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

Definition at line 711 of file String.h.

References RCPP_STRING_DEBUG.

5.2.3.420 wrap_extra_steps()

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

Definition at line 28 of file wrap_extra_steps.h.

5.2.3.421 wrap_extra_steps< Rcpp::Date >()

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

Definition at line 151 of file Date.h.

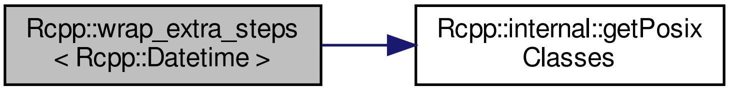
5.2.3.422 wrap_extra_steps< Rcpp::Datetime >()

```
template<>
SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (
             SEXP  )  [inline]
```

Definition at line 38 of file Datetime.h.

References Rcpp::internal::getPosixClasses().

Here is the call graph for this function:



5.2.4 Variable Documentation

5.2.4.1 _

```
internal::NamedPlaceHolder Rcpp::_  [static]
```

Definition at line 64 of file Named.h.

Referenced by Rcpp::attributes::checkRSignature(), Rcpp::exception::copy_stack_trace_to_r(), Rcpp::Module::invoke(), new_env(), shush_about_NA(), and sourceCppContext().

5.2.4.2 BindingPolicy< Environment_Impl< StoragePolicy > >

```
public Rcpp::BindingPolicy< Environment_Impl< StoragePolicy > >
```

Initial value:

```
{
    private:
        inline SEXP as_environment(SEXP x){
            if( Rf_isEnvironment(x) ) return x ;
            SEXP asEnvironmentSym = Rf_install("as.environment");
            try {
                Shield<SEXP> call(Rf_lang2(asEnvironmentSym, x));
                return Rcpp_fast_eval(call, R_GlobalEnv);
            } catch( const eval_error& ex) {
                const char* fmt = "Cannot convert object to an environment: "
                                  "[type=%s; target=ENVSXP].";
                throw not_compatible(fmt, Rf_type2char(TYPEOF(x)));
            }
        }
    public:
        Environment_Impl(){
            Storage::set__(R_GlobalEnv) ;
        }
```

Definition at line 29 of file Environment.h.

5.2.4.3 current_scope

```
Module* Rcpp::current_scope  [static]
```

Definition at line 224 of file module.cpp.

Referenced by getCurrentScope(), and setCurrentScope().

5.2.4.4 DottedPairImpl< DottedPair_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< DottedPair_Impl< StoragePolicy > >
```

Initial value:

```
{
public:
        DottedPair_Impl(){}
        DottedPair_Impl(SEXP x) {
            Storage::set__(x) ;
        }
        void update(SEXP){}
}
```

Definition at line 29 of file DottedPair.h.

5.2.4.5 DottedPairImpl< Formula_Impl< StoragePolicy > >

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

Initial value:

```
{
    public:
        Formula_Impl(){}
```

Definition at line 32 of file Formula.h.

5.2.4.6 DottedPairImpl< Language_Impl< StoragePolicy > >

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

Initial value:

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

Definition at line 33 of file Language.h.

5.2.4.7 DottedPairImpl< Pairlist_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< Pairlist_Impl< StoragePolicy > >
```

Initial value:

```
{
    public:
        typedef typename DottedPairProxyPolicy<Pairlist_Impl>::DottedPairProxy Proxy
```

Definition at line 33 of file Pairlist.h.

5.2.4.8 DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >

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

Definition at line 28 of file DottedPair.h.

5.2.4.9 DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >

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

Definition at line 31 of file Formula.h.

5.2.4.10 DottedPairProxyPolicy< Language_Impl< StoragePolicy > >

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

Definition at line 32 of file Language.h.

5.2.4.11 DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >

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

Definition at line 32 of file Pairlist.h.

5.2.4.12 DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > >
```

Initial value:

```
{
    public:
        typedef typename DottedPairProxyPolicy<StretchyList_Impl>::DottedPairProxy Proxy
```

Definition at line 34 of file StretchyList.h.

5.2.4.13 FieldProxyPolicy< Reference_Impl< StoragePolicy > >

```
public Rcpp::FieldProxyPolicy< Reference_Impl< StoragePolicy > >
```

Initial value:

```
{
    public:
        Reference_Impl() {}

        Reference_Impl(SEXP x) {
            Storage::set__(x) ;
        }
        Reference_Impl& operator=( SEXP other ) {
            Storage::set__(other) ;
            return *this ;
        }

        Reference_Impl( const std::string& klass ) {
            SEXP newSym = Rf_install("new");
            Shield<SEXP> str(Rf_mkString(klass.c_str()));
            Shield<SEXP> call(Rf_lang2(newSym, str));
            Storage::set__( Rcpp_fast_eval( call , Rcpp::internal::get_Rcpp_namespace()) );
        }
        void update( SEXP x){
            if( ! ::Rf_isS4(x) ) throw not_reference();
        }
    }
```

Definition at line 31 of file Reference.h.

5.2.4.14 gmt

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

Definition at line 358 of file date.cpp.

Referenced by gmtload().

5.2.4.15 gmt_is_set

```
int Rcpp::gmt_is_set  [static]
```

Definition at line 429 of file date.cpp.

Referenced by gmtsub().

5.2.4.16 gmtmem

```
struct state Rcpp::gmtmem  [static]
```

Definition at line 429 of file date.cpp.

5.2.4.17 mon_lengths

```
const int Rcpp::mon_lengths[2][MONSPERYEAR]  [static]
```

Initial value:

```
= {
        { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },
        { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }
    }
```

Definition at line 420 of file date.cpp.

Referenced by timesub(), and transtime().

5.2.4.18 NA

```
Na_Proxy Rcpp::NA  [static]
```

Definition at line 52 of file Na_Proxy.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression(), Rcpp::algorithm::max(), Rcpp::algorithm::min(), Rcpp::algorithm::helpers::rtype< T >::NA(), operator!=(), operator<(), operator<=(), operator==(), operator>(), operator>=(), Rcpp::algorithm::prod(), sapply(), shush_about_NA(), and Rcpp::algorithm::sum().

5.2.4.19 Rcerr

```
Rostream< false > & Rcpp::Rcerr = Rcpp_cerr_get()  [static]
```

Definition at line 90 of file Rstreambuf.h.

5.2.4.20 Rcout

```
Rostream< true > & Rcpp::Rcout = Rcpp_cout_get()  [static]
```

Definition at line 89 of file Rstreambuf.h.

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

5.2.4.21 Rcpp_precious

```
SEXP Rcpp::Rcpp_precious = R_NilValue  [static]
```

Definition at line 92 of file barrier.cpp.

5.2.4.22 time_t_max

```
time_t const Rcpp::time_t_max = MAXVAL(time_t, TYPE_BIT(time_t))  [static]
```

Definition at line 162 of file date.cpp.

Referenced by increment_overflow_time().

5.2.4.23 time_t_min

```
time_t const Rcpp::time_t_min = MINVAL(time_t, TYPE_BIT(time_t))  [static]
```

Definition at line 161 of file date.cpp.

Referenced by increment_overflow_time().

5.2.4.24 tm

```
struct tm Rcpp::tm  [static]
```

Definition at line 429 of file date.cpp.

Referenced by Rcpp::Datetime::format(), Rcpp::Date::format(), gmtime_(), gmtsub(), and mktime00().

5.2.4.25 year_lengths

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

Initial value:

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

Definition at line 425 of file date.cpp.

Referenced by timesub(), and tzparse().

5.3 Rcpp::algorithm Namespace Reference

Namespaces

- helpers

Functions

- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type sum (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type sum_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type prod (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type prod_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type max (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type max_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type min (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type min_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value &&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::type, double >::value, double >::type mean (InputIterator begin, InputIterator end)
- template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value &&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↩
::type, int >::value, double >::type mean (InputIterator begin, InputIterator end)
- template<typename InputIterator , typename OutputIterator >
void log (InputIterator begin, InputIterator end, OutputIterator out)
- template<typename InputIterator , typename OutputIterator >
void exp (InputIterator begin, InputIterator end, OutputIterator out)
- template<typename InputIterator , typename OutputIterator >
void sqrt (InputIterator begin, InputIterator end, OutputIterator out)

5.3.1 Function Documentation

5.3.1.1 exp()

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

Definition at line 474 of file algorithm.h.

Referenced by Rcpp::internal::complex__exp(), Rcpp::stats::d_exp_0(), Rcpp::stats::dlnorm_0(), Rcpp::stats::dlnorm_1(), Rcpp::stats::dlogis_0(), Rcpp::stats::dlogis_1(), Rcpp::stats::dnorm_0(), Rcpp::stats::dnorm_1(), Rcpp::stats::dweibull_1(), Rcpp::stats::LNormGenerator::operator()(), Rcpp::stats::LNormGenerator_1::operator()(), Rcpp::stats::LNormGenerator_0::operator()(), Rcpp::algorithm::helpers::exp::operator()(), Rcpp::stats::p_exp_0(), Rcpp::stats::plogis_0(), Rcpp::stats::plogis_1(), Rcpp::stats::pweibull_1(), Rcpp::stats::qlnorm_0(), Rcpp::stats::qlnorm_1(), Rcpp::stats::qlogis_0(), Rcpp::stats::qlogis_1(), and Rcpp::rlnorm().

5.3.1.2 log()

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

Definition at line 469 of file algorithm.h.

Referenced by Rcpp::internal::complex__asin(), Rcpp::internal::complex__atan(), Rcpp::internal::complex__log(), Rcpp::dexp(), Rcpp::stats::dgamma_1(), Rcpp::stats::dlnorm_0(), Rcpp::stats::dlnorm_1(), Rcpp::stats::dlogis_0(), Rcpp::stats::dlogis_1(), Rcpp::stats::dweibull_1(), Rcpp::stats::LogisGenerator::operator()(), Rcpp::stats::LogisGenerator_1::operator()(), Rcpp::stats::LogisGenerator_0::operator()(), Rcpp::stats::WeibullGenerator::operator()(), Rcpp::stats::WeibullGenerator__scale1::operator()(), Rcpp::algorithm::helpers::log::operator()(), Rcpp::stats::p_exp_0(), Rcpp::pexp(), Rcpp::stats::plnorm_0(), Rcpp::stats::plnorm_1(), Rcpp::stats::pweibull_1(), Rcpp::qexp(), Rcpp::stats::qlogis_0(), and Rcpp::stats::qlogis_1().

5.3.1.3 max()

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

Definition at line 324 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices(), max_nona(), and Rcpp::operator<<().

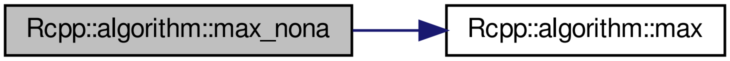
5.3.1.4 max_nona()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↩
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↩
::value_type >::type >::type Rcpp::algorithm::max_nona (
             InputIterator begin,
             InputIterator end )
```

Definition at line 349 of file algorithm.h.

References max().

Here is the call graph for this function:



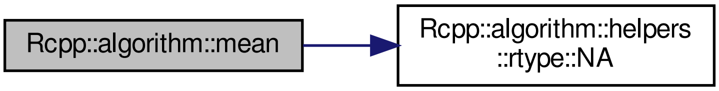
5.3.1.5 mean() [1/2]

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

Definition at line 417 of file algorithm.h.

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

Here is the call graph for this function:



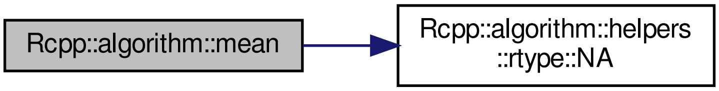
5.3.1.6 mean() [2/2]

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

Definition at line 444 of file algorithm.h.

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

Here is the call graph for this function:



5.3.1.7 min()

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

Definition at line 370 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), tinyformat::detail::formatTruncated(), min_nona(), Rcpp::operator<<(), and PiLeibniz().

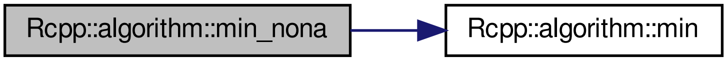
5.3.1.8 min_nona()

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

Definition at line 395 of file algorithm.h.

References min().

Here is the call graph for this function:



5.3.1.9 prod()

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

Definition at line 278 of file algorithm.h.

References Rcpp::NA.

5.3.1.10 prod_nona()

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

Definition at line 303 of file algorithm.h.

5.3.1.11 sqrt()

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

Definition at line 479 of file algorithm.h.

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

5.3.1.12 sum()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↩
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↩
::value_type >::type >::type Rcpp::algorithm::sum (
             InputIterator begin,
             InputIterator end )
```

Definition at line 232 of file algorithm.h.

References Rcpp::NA.

5.3.1.13 sum_nona()

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

Definition at line 257 of file algorithm.h.

5.4 Rcpp::algorithm::helpers Namespace Reference

Classes

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

- struct ctype_helper< sizeof(CTYPE_FLOAT) >
- struct ctype_helper< sizeof(CTYPE_DOUBLE) >
- struct ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >
- struct ctype_helper< sizeof(CTYPE_STRING) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >
- struct ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >
- struct ctype
- struct decays_to_ctype
- struct rtype_helper
- struct rtype_helper< double >
- struct rtype_helper< int >
- struct rtype
- struct log
- struct exp
- struct sqrt

5.5 Rcpp::attributes Namespace Reference

Classes

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

Functions

- bool removeFile (const std::string &path)
- void createDirectory (const std::string &path)
- bool isWhitespace (char ch)
- void trimWhitespace (std::string ∗pStr)
- void stripTrailingLineComments (std::string ∗pStr)
- void stripQuotes (std::string ∗pStr)
- bool isQuoted (const std::string &str)

- bool endsWith (const std::string &str, const std::string &suffix)
- void showWarning (const std::string &msg)
- bool isRoxygenCpp (const std::string &str)
- std::ostream & operator<< (std::ostream &os, const Type &type)
- std::ostream & operator<< (std::ostream &os, const Argument &argument)
- std::ostream & operator<< (std::ostream &os, const Function &function)
- std::ostream & operator<< (std::ostream &os, const Param &param)
- std::ostream & operator<< (std::ostream &os, const Attribute &attribute)
- std::string generateRArgList (const Function &function)
- bool checkRSignature (const Function &function, std::string args)
- void initializeGlobals (std::ostream &ostr)
- void generateCpp (std::ostream &ostr, const SourceFileAttributes &attributes, bool includePrototype, bool cppInterface, 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 kExportInvisible = "invisible"
- const char ∗const kExportSignature = "signature"
- const char ∗const kInitAttribute = "init"
- const char ∗const kDependsAttribute = "depends"
- const char ∗const kPluginsAttribute = "plugins"
- const char ∗const kInterfacesAttribute = "interfaces"
- const char ∗const kInterfaceR = "r"
- const char ∗const kInterfaceCpp = "cpp"
- const char ∗const kParamValueFalse = "false"
- const char ∗const kParamValueTrue = "true"
- const char ∗const kParamValueFALSE = "FALSE"
- const char ∗const kParamValueTRUE = "TRUE"
- const char ∗const kParamBlockStart = "{;"
- const char ∗const kParamBlockEnd = "}"

5.5.1 Function Documentation

5.5.1.1 checkRSignature()

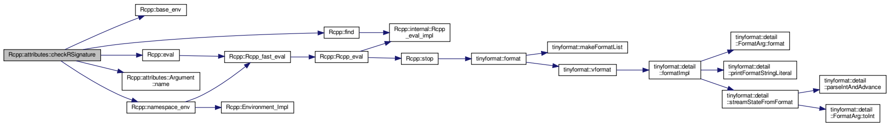
```
bool Rcpp::attributes::checkRSignature (
            const Function & function,
            std::string args )
```

Definition at line 2803 of file attributes.cpp.

References Rcpp::_, Rcpp::base_env(), Rcpp::eval(), Rcpp::find(), Rcpp::attributes::Argument::name(), and Rcpp↩
::namespace_env().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::checkRSignature
Rcpp::base_env
Rcpp::find
Rcpp::internal::Rcpp_eval_impl
Rcpp::eval
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::stop
Rcpp::attributes::Argument::name
Rcpp::namespace_env
Rcpp::Environment_Impl
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

5.5.1.2 createDirectory()

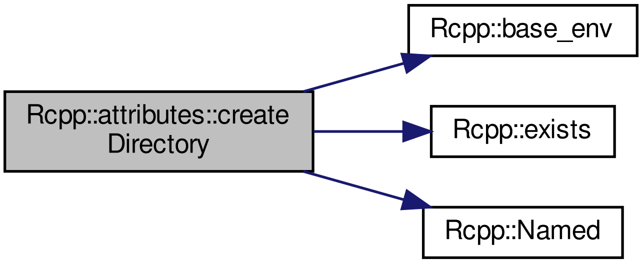
```
void Rcpp::attributes::createDirectory (
            const std::string & path )
```

Definition at line 3014 of file attributes.cpp.

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

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), and Rcpp::attributes::CppPackageInclude↩
Generator::commit().

Here is the call graph for this function:



5.5.1.3 endsWith()

```
bool Rcpp::attributes::endsWith (
            const std::string & str,
            const std::string & suffix )
```

Definition at line 3109 of file attributes.cpp.

Referenced by compileAttributes().

5.5.1.4 generateCpp()

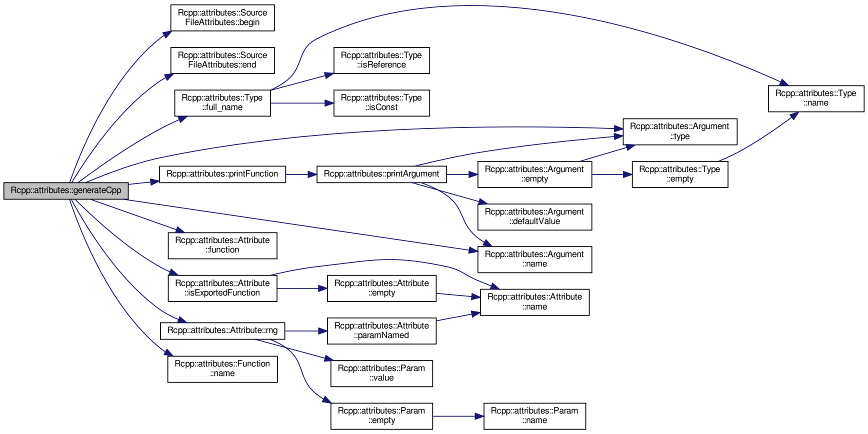
```
void Rcpp::attributes::generateCpp (
            std::ostream & ostr,
            const SourceFileAttributes & attributes,
            bool includePrototype,
            bool cppInterface,
            const std::string & contextId )
```

Definition at line 2843 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::Type::full_name(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::Attribute::isExportedFunction(), Rcpp::attributes::Argument::name(), Rcpp::attributes::Function::name(), printFunction(), Rcpp::attributes::Attribute::rng(), and Rcpp::attributes::Argument::type().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::Source FileAttributes::begin
Rcpp::attributes::Source FileAttributes::end
Rcpp::attributes::Type ::isReference
Rcpp::attributes::Type ::full_name
Rcpp::attributes::Type ::isConst
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::type
Rcpp::attributes::generateCpp
Rcpp::attributes::printFunction
Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Type ::empty
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Attribute ::function
Rcpp::attributes::Argument ::name
Rcpp::attributes::Attribute ::isExportedFunction
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute::rng
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Function ::name
Rcpp::attributes::Param ::value
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name

5.5.1.5 generateRArgList()

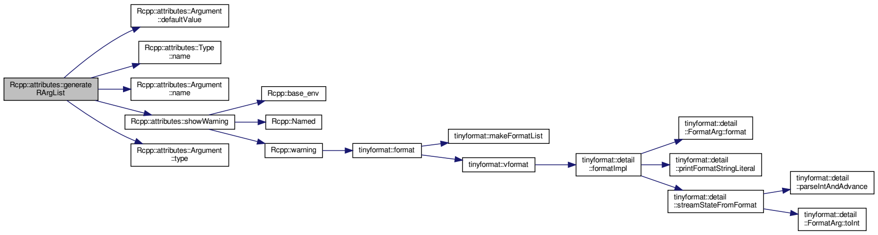
```
std::string Rcpp::attributes::generateRArgList (
            const Function & function )
```

Definition at line 2778 of file attributes.cpp.

References Rcpp::attributes::Argument::defaultValue(), Rcpp::attributes::Type::name(), Rcpp::attributes::Argument↩ ::name(), showWarning(), and Rcpp::attributes::Argument::type().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::generate RArgList
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::name
Rcpp::attributes::showWarning
Rcpp::attributes::Argument ::type
Rcpp::base_env
Rcpp::Named
Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.5.1.6 initializeGlobals()

```
void Rcpp::attributes::initializeGlobals (
            std::ostream & ostr )
```

Definition at line 2832 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::commit().

5.5.1.7 isQuoted()

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

Definition at line 3101 of file attributes.cpp.

5.5.1.8 isRoxygenCpp()

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

Definition at line 3121 of file attributes.cpp.

References kWhitespaceChars.

Referenced by stripTrailingLineComments().

5.5.1.9 isWhitespace()

```
bool Rcpp::attributes::isWhitespace (
             char ch )
```

Definition at line 3025 of file attributes.cpp.

References kWhitespaceChars.

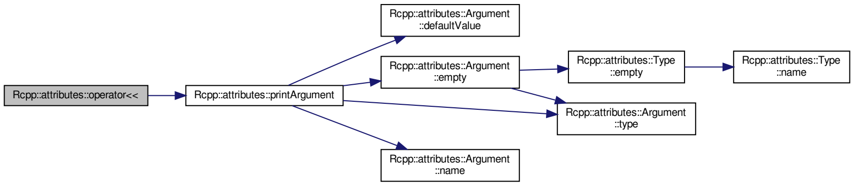
5.5.1.10 operator<<() [1/5]

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

Definition at line 1119 of file attributes.cpp.

References printArgument().

Here is the call graph for this function:


Rcpp::attributes::operator<<
Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::type
Rcpp::attributes::Argument ::name

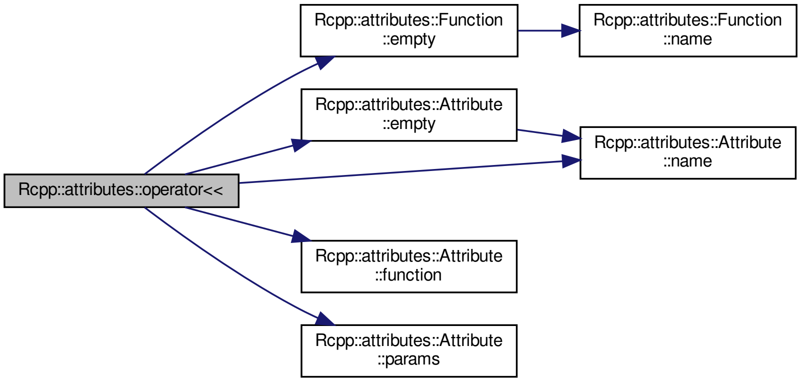
5.5.1.11 operator<<() [2/5]

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

Definition at line 1163 of file attributes.cpp.

References Rcpp::attributes::Function::empty(), Rcpp::attributes::Attribute::empty(), Rcpp::attributes::Attribute↩ ::function(), Rcpp::attributes::Attribute::name(), and Rcpp::attributes::Attribute::params().

Here is the call graph for this function:


Rcpp::attributes::operator<<
Rcpp::attributes::Function ::empty
Rcpp::attributes::Function ::name
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute ::function
Rcpp::attributes::Attribute ::params

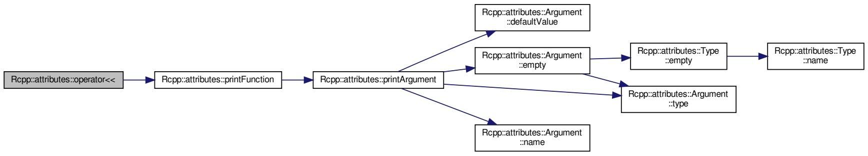
5.5.1.12 operator<<() [3/5]

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

Definition at line 1147 of file attributes.cpp.

References printFunction().

Here is the call graph for this function:


Rcpp::attributes::operator<<
Rcpp::attributes::printFunction
Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Argument ::type
Rcpp::attributes::Argument ::name
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name

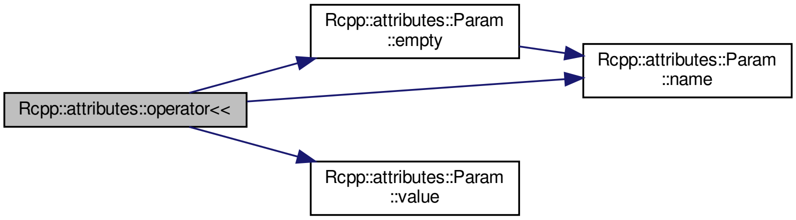
5.5.1.13 operator<<() [4/5]

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

Definition at line 1153 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::Param::name(), and Rcpp::attributes::Param::value().

Here is the call graph for this function:


Rcpp::attributes::operator<<
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Param ::value

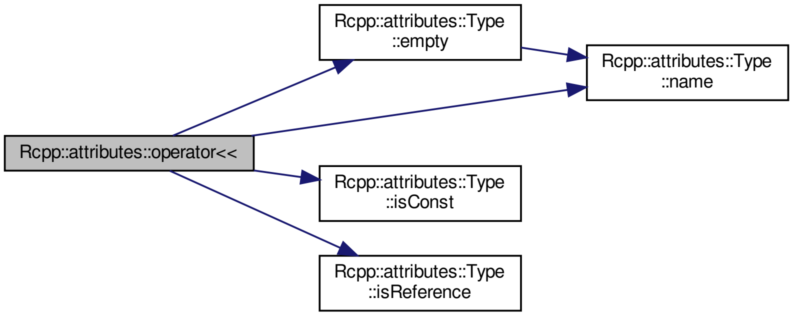
5.5.1.14 operator<<() [5/5]

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

Definition at line 1092 of file attributes.cpp.

References Rcpp::attributes::Type::empty(), Rcpp::attributes::Type::isConst(), Rcpp::attributes::Type::isReference(), and Rcpp::attributes::Type::name().

Here is the call graph for this function:


Rcpp::attributes::operator<<
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name
Rcpp::attributes::Type ::isConst
Rcpp::attributes::Type ::isReference

5.5.1.15 printArgument()

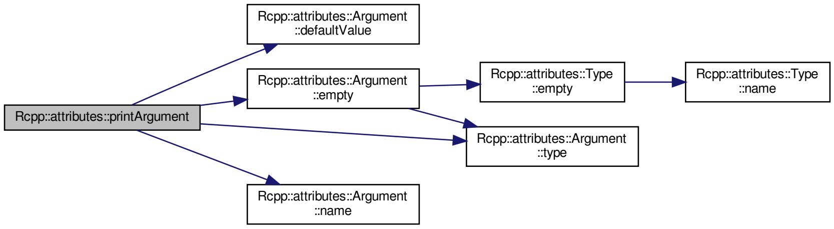
```
void Rcpp::attributes::printArgument (
             std::ostream & os,
             const Argument & argument,
             bool printDefault = true )
```

Definition at line 1104 of file attributes.cpp.

References Rcpp::attributes::Argument::defaultValue(), Rcpp::attributes::Argument::empty(), Rcpp::attributes::↩ Argument::name(), and Rcpp::attributes::Argument::type().

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

Here is the call graph for this function:


Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Argument ::type
Rcpp::attributes::Argument ::name
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name

5.5.1.16 printFunction()

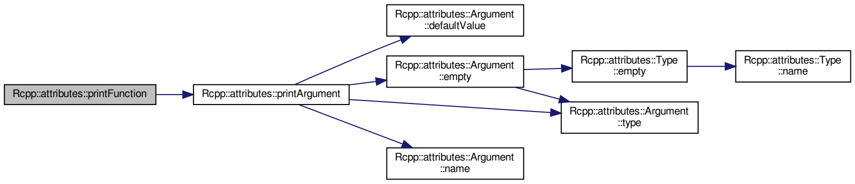
```
void Rcpp::attributes::printFunction (
             std::ostream & os,
             const Function & function,
             bool printArgDefaults = true )
```

Definition at line 1125 of file attributes.cpp.

References printArgument().

Referenced by generateCpp(), operator<<(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::attributes::printFunction
Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Argument ::type
Rcpp::attributes::Argument ::name
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name

5.5.1.17 removeFile()

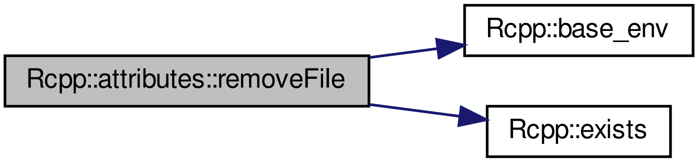
```
bool Rcpp::attributes::removeFile (
            const std::string & path )
```

Definition at line 3002 of file attributes.cpp.

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

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

Here is the call graph for this function:



5.5.1.18 showWarning()

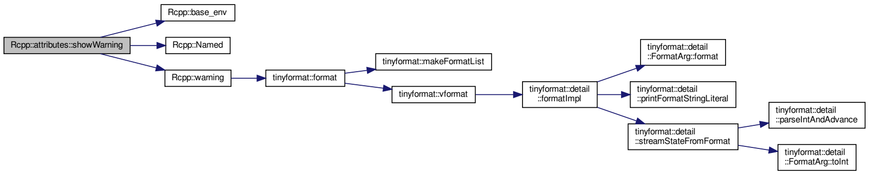
```
void Rcpp::attributes::showWarning (
            const std::string & msg )
```

Definition at line 3116 of file attributes.cpp.

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

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

Here is the call graph for this function:


Rcpp::attributes::showWarning
Rcpp::base_env
Rcpp::Named
Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.5.1.19 stripQuotes()

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

Definition at line 3092 of file attributes.cpp.

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

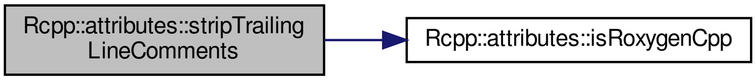
5.5.1.20 stripTrailingLineComments()

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

Definition at line 3031 of file attributes.cpp.

References isRoxygenCpp(), and kWhitespaceChars.

Here is the call graph for this function:


Rcpp::attributes::stripTrailing
LineComments
Rcpp::attributes::isRoxygenCpp

5.5.1.21 trimWhitespace()

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

Definition at line 3075 of file attributes.cpp.

References kWhitespaceChars.

Referenced by Rcpp::attributes::Attribute::customRSignature(), and Rcpp::attributes::Param::Param().

5.5.2 Variable Documentation

5.5.2.1 kDependsAttribute

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

Definition at line 159 of file attributes.cpp.

Referenced by compileAttributes().

5.5.2.2 kExportAttribute

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

Definition at line 153 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::isExportedFunction().

5.5.2.3 kExportInvisible

```
const char* const Rcpp::attributes::kExportInvisible = "invisible"
```

Definition at line 156 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::invisible().

5.5.2.4 kExportName

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

Definition at line 154 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::exportedName().

5.5.2.5 kExportRng

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

Definition at line 155 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::rng().

5.5.2.6 kExportSignature

```
const char* const Rcpp::attributes::kExportSignature = "signature"
```

Definition at line 157 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::customRSignature(), and Rcpp::attributes::RExportsGenerator::doWriteFunctions().

5.5.2.7 kInitAttribute

```
const char* const Rcpp::attributes::kInitAttribute = "init"
```

Definition at line 158 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

5.5.2.8 kInterfaceCpp

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

Definition at line 163 of file attributes.cpp.

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

5.5.2.9 kInterfaceR

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

Definition at line 162 of file attributes.cpp.

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

5.5.2.10 kInterfacesAttribute

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

Definition at line 161 of file attributes.cpp.

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

5.5.2.11 kParamBlockEnd

```
const char* const Rcpp::attributes::kParamBlockEnd = "}"
```

Definition at line 169 of file attributes.cpp.

5.5.2.12 kParamBlockStart

```
const char* const Rcpp::attributes::kParamBlockStart = "{;"
```

Definition at line 168 of file attributes.cpp.

5.5.2.13 kParamValueFalse

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

Definition at line 164 of file attributes.cpp.

5.5.2.14 kParamValueFALSE

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

Definition at line 166 of file attributes.cpp.

5.5.2.15 kParamValueTrue

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

Definition at line 165 of file attributes.cpp.

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

5.5.2.16 kParamValueTRUE

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

Definition at line 167 of file attributes.cpp.

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

5.5.2.17 kPluginsAttribute

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

Definition at line 160 of file attributes.cpp.

5.5.2.18 kWhitespaceChars

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

Definition at line 3022 of file attributes.cpp.

Referenced by isRoxygenCpp(), isWhitespace(), stripTrailingLineComments(), and trimWhitespace().

5.6 Rcpp::internal Namespace Reference

internal implementation details

Namespaces

- debug

Classes

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

Functions

- template<> double caster< Rcpp::Date, double > (Rcpp::Date from)
- template<> Rcpp::Date caster< double, Rcpp::Date > (double from)
- template<> double caster< Rcpp::Datetime, double > (Rcpp::Datetime from)
- template<> Rcpp::Datetime caster< double, Rcpp::Datetime > (double from)
- void ∗ as_module_object_internal (SEXP obj)
- template<typename InputIterator , typename value_type >
  void export_range__dispatch (SEXP x, InputIterator first, ::Rcpp::traits::r_type_generic_tag)
- bool is_atomic (SEXP x)
- bool is_matrix (SEXP x)
- template<> bool is__simple< int > (SEXP x)
- template<> bool is__simple< double > (SEXP x)
- template<> bool is__simple< bool > (SEXP x)
- template<> bool is__simple< std::string > (SEXP x)
- template<> bool is__simple< String > (SEXP x)
- template<> bool is__simple< Rcomplex > (SEXP x)
- template<> bool is__simple< CharacterVector > (SEXP x)
- template<> bool is__simple< CharacterMatrix > (SEXP x)
- template<> bool is__simple< RObject > (SEXP)
- template<> bool is__simple< IntegerVector > (SEXP x)
- template<> bool is__simple< ComplexVector > (SEXP x)
- template<> bool is__simple< RawVector > (SEXP x)
- template<> bool is__simple< NumericVector > (SEXP x)
- template<> bool is__simple< LogicalVector > (SEXP x)
- template<> bool is__simple< Language > (SEXP x)
- template<> bool is__simple< DottedPair > (SEXP x)
- template<> bool is__simple< List > (SEXP x)
- template<> bool is__simple< IntegerMatrix > (SEXP x)
- template<> bool is__simple< ComplexMatrix > (SEXP x)

- template<> bool is__simple< RawMatrix > (SEXP x)
- template<> bool is__simple< NumericMatrix > (SEXP x)
- template<> bool is__simple< LogicalMatrix > (SEXP x)
- template<> bool is__simple< GenericMatrix > (SEXP x)
- template<> bool is__simple< DataFrame > (SEXP x)
- template<> bool is__simple< WeakReference > (SEXP x)
- template<> bool is__simple< Symbol > (SEXP x)
- template<> bool is__simple< S4 > (SEXP x)
- template<> bool is__simple< Reference > (SEXP x)
- template<> bool is__simple< Promise > (SEXP x)
- template<> bool is__simple< Pairlist > (SEXP x)
- template<> bool is__simple< Function > (SEXP x)
- template<> bool is__simple< Environment > (SEXP x)
- template<> bool is__simple< Formula > (SEXP x)
- template<> bool is__simple< Date > (SEXP x)
- template<> bool is__simple< Datetime > (SEXP x)
- template<> bool is__simple< DateVector > (SEXP x)
- template<> bool is__simple< DatetimeVector > (SEXP x)
- bool is_module_object_internal (SEXP obj, const char ∗clazz)
- template<typename T >
  bool is__module__object (SEXP x)
- SEXP Rcpp_eval_impl (SEXP expr, SEXP env)
- template<typename T >
  SEXP wrap_range_sugar_expression (const T &object, Rcpp::traits::true_type)
- template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
  SEXP range_wrap_dispatch___impl__pair (InputIterator first, InputIterator last, Rcpp::traits::true_type)
- template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
  SEXP range_wrap_dispatch___impl__pair (InputIterator first, InputIterator last, Rcpp::traits::false_type)
- template<typename T >
  T primitive_as (SEXP x)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_primitive_tag)
- const char ∗ check_single_string (SEXP x)
- template<typename T >
  T as_string (SEXP x, Rcpp::traits::true_type)
- template<typename T >
  T as_string (SEXP x, Rcpp::traits::false_type)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_string_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_RcppString_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_generic_tag)
- template<typename T >
  object< T > as_module_object (SEXP x)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_module_object_const_pointer_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_module_object_pointer_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_module_object_tag)

- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_module_object_reference_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_module_object_const_reference_tag)
- template<typename T >
  T as (SEXP x, ::Rcpp::traits::r_type_enum_tag)
- SEXP empty_data_frame ()
- SEXP getPosixClasses ()
- SEXP new_posixt_object (double d)
- SEXP new_date_object (double d)
- SEXP longjumpSentinel (SEXP token)
- bool isLongjumpSentinel (SEXP x)
- SEXP getLongjumpToken (SEXP sentinel)
- void resumeJump (SEXP token)
- SEXP nth (SEXP s, int n)
- bool is_Rcpp_eval_call (SEXP expr)
- template<typename T >
  SEXP grow__dispatch (::Rcpp::traits::false_type, const T &head, SEXP tail)
- template<typename T >
  SEXP grow__dispatch (::Rcpp::traits::true_type, const T &head, SEXP tail)
- SEXP interruptedError ()
- template<typename T >
  bool is__simple (SEXP x)
- template<typename T >
  bool is__dispatch (SEXP x, Rcpp::traits::false_type)
- template<typename T >
  bool is__dispatch (SEXP x, Rcpp::traits::true_type)
- template<typename FROM , typename TO >
  std::string get_converter_name (const char ∗from, const char ∗to)
- template<typename Class >
  SEXP make_new_object (Class ∗ptr)
- SEXP convert_using_rfunction (SEXP x, const char ∗const fun)
- template<int TARGET>
  SEXP r_true_cast (SEXP x)
- template<int RTYPE>
  SEXP basic_cast (SEXP x)
- template<> SEXP r_true_cast< INTSXP > (SEXP x)
- template<> SEXP r_true_cast< REALSXP > (SEXP x)
- template<> SEXP r_true_cast< RAWSXP > (SEXP x)
- template<> SEXP r_true_cast< CPLXSXP > (SEXP x)
- template<> SEXP r_true_cast< LGLSXP > (SEXP x)
- template<> SEXP r_true_cast< STRSXP > (SEXP x)
- template<> SEXP r_true_cast< VECSXP > (SEXP x)
- template<> SEXP r_true_cast< EXPRSXP > (SEXP x)
- template<> SEXP r_true_cast< LISTSXP > (SEXP x)
- template<> SEXP r_true_cast< LANGSXP > (SEXP x)
- attribute_hidden unsigned long enterRNGScope ()
- attribute_hidden unsigned long exitRNGScope ()
- attribute_hidden unsigned long beginSuspendRNGSynchronization ()
- attribute_hidden unsigned long endSuspendRNGSynchronization ()
- attribute_hidden char ∗ get_string_buffer ()

- attribute_hidden SEXP get_Rcpp_namespace ()
- template<> SEXP make_charsexp< Rcpp::String > (const Rcpp::String &s)
- double complex__Re (Rcomplex x)
- double complex__Im (Rcomplex x)
- double complex__Mod (Rcomplex x)
- Rcomplex complex__Conj (Rcomplex x)
- double complex__Arg (Rcomplex x)
- Rcomplex complex__exp (Rcomplex x)
- Rcomplex complex__log (Rcomplex x)
- Rcomplex complex__sqrt (Rcomplex z)
- Rcomplex complex__cos (Rcomplex z)
- Rcomplex complex__cosh (Rcomplex z)
- Rcomplex complex__sin (Rcomplex z)
- Rcomplex complex__tan (Rcomplex z)
- Rcomplex complex__asin (Rcomplex z)
- Rcomplex complex__acos (Rcomplex z)
- Rcomplex complex__atan (Rcomplex z)
- Rcomplex complex__acosh (Rcomplex z)
- Rcomplex complex__asinh (Rcomplex z)
- Rcomplex complex__atanh (Rcomplex z)
- Rcomplex complex__sinh (Rcomplex z)
- Rcomplex complex__tanh (Rcomplex z)
- double factorial (double x)
- double lfactorial (double x)
- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > as_vector__impl (MatrixBase< RTYPE, NA, T > &t, Rcpp::traits::false_type)
- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > as_vector__impl (MatrixBase< RTYPE, NA, T > &t, Rcpp::traits::true_type)
- int get_line (int index, int nr)
- int get_column (int index, int nr)
- int get_column (int index, int nr, int i)
- void maybeJump (void ∗unwind_data, Rboolean jump)
- template<int RT>
  bool operator< (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool operator> (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool operator>= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool operator<= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & operator<< (std::ostream &os, const const_string_proxy< STRSXP, StoragePolicy > &proxy)
- template<template< class > class StoragePolicy>
  std::string operator+ (const std::string &x, const const_string_proxy< STRSXP, StoragePolicy > &y)
- template<int RT>
  bool operator< (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool operator> (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool operator>= (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool operator<= (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)

- std::ostream & operator<< (std::ostream &os, const string_proxy< STRSXP > &proxy)
- std::string operator+ (const std::string &x, const string_proxy< STRSXP > &y)
- template<int RTYPE>
  SEXP vector_from_string (const std::string &st)
- template<int RTYPE>
  SEXP vector_from_string_expr (const std::string &code)
- template<> SEXP vector_from_string< EXPRSXP > (const std::string &st)

Variables

- int rngSynchronizationSuspended = 0

5.6.1 Detailed Description

internal implementation details

5.6.2 Function Documentation

5.6.2.1 as() [1/10]

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

handling enums by converting to int first

Definition at line 129 of file as.h.

5.6.2.2 as() [2/10]

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

Definition at line 85 of file as.h.

References DEMANGLE, and RCPP_DEBUG_1.

5.6.2.3 as() [3/10]

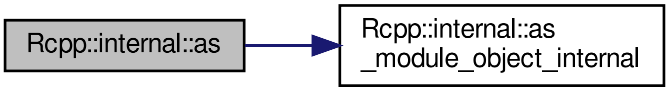
```
template<typename T >
T Rcpp::internal::as (
             SEXP x,
             ::Rcpp::traits::r_type_module_object_const_pointer_tag  )
```

handling object<T>

Definition at line 99 of file as.h.

References as_module_object_internal().

Here is the call graph for this function:



5.6.2.4 as() [4/10]

```
template<typename T >
T Rcpp::internal::as (
             SEXP x,
             ::Rcpp::traits::r_type_module_object_const_reference_tag  )
```

handling T such that T is a reference of a class handled by a module

Definition at line 122 of file as.h.

5.6.2.5 as() [5/10]

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

Definition at line 104 of file as.h.

5.6.2.6 as() [6/10]

```
template<typename T >
T Rcpp::internal::as (
             SEXP x,
             ::Rcpp::traits::r_type_module_object_reference_tag )
```

handling T such that T is a reference of a class handled by a module

Definition at line 115 of file as.h.

5.6.2.7 as() [7/10]

```
template<typename T >
T Rcpp::internal::as (
             SEXP x,
             ::Rcpp::traits::r_type_module_object_tag )
```

handling T such that T is exposed by a module

Definition at line 109 of file as.h.

5.6.2.8 as() [8/10]

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

Definition at line 43 of file as.h.

5.6.2.9 as() [9/10]

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

Definition at line 74 of file as.h.

5.6.2.10 as() [10/10]

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

Definition at line 70 of file as.h.

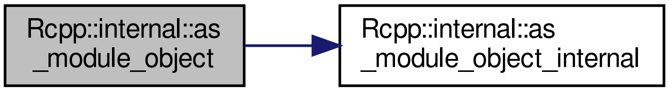
5.6.2.11 as_module_object()

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

Definition at line 94 of file as.h.

References as_module_object_internal().

Here is the call graph for this function:



5.6.2.12 as_module_object_internal()

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

Definition at line 24 of file as.h.

Referenced by as(), and as_module_object().

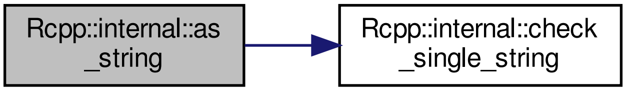
5.6.2.13 as_string() [1/2]

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

Definition at line 66 of file as.h.

References check_single_string().

Here is the call graph for this function:



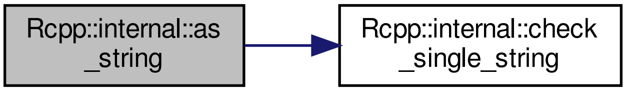
5.6.2.14 as_string() [2/2]

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

Definition at line 61 of file as.h.

References check_single_string().

Here is the call graph for this function:



5.6.2.15 as_vector__impl() [1/2]

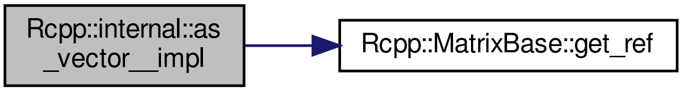
```
template<int RTYPE, bool NA, typename T >
Rcpp::Vector<RTYPE> Rcpp::internal::as_vector__impl (
            MatrixBase< RTYPE, NA, T > & t,
            Rcpp::traits::false_type  )  [inline]
```

Definition at line 30 of file as_vector.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref().

Referenced by Rcpp::as_vector().

Here is the call graph for this function:



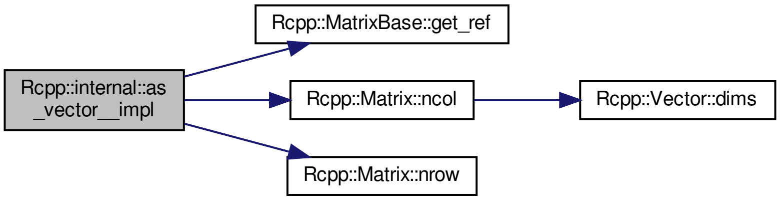
5.6.2.16 as_vector__impl() [2/2]

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

Definition at line 44 of file as_vector.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:



5.6.2.17 basic_cast()

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

Definition at line 59 of file r_cast.h.

5.6.2.18 beginSuspendRNGSynchronization()

```
unsigned long Rcpp::internal::beginSuspendRNGSynchronization ( )  [inline]
```

Definition at line 103 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::SuspendRNGSynchronizationScope::SuspendRNGSynchronization↩ Scope().

5.6.2.19 caster< double, Rcpp::Date >()

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

Definition at line 146 of file Date.h.

5.6.2.20 caster< double, Rcpp::Datetime >()

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

Definition at line 131 of file Datetime.h.

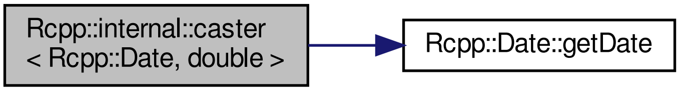
5.6.2.21 caster< Rcpp::Date, double >()

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

Definition at line 143 of file Date.h.

References Rcpp::Date::getDate().

Here is the call graph for this function:



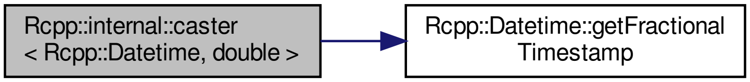
5.6.2.22 caster< Rcpp::Datetime, double >()

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

Definition at line 128 of file Datetime.h.

References Rcpp::Datetime::getFractionalTimestamp().

Here is the call graph for this function:



5.6.2.23 check_single_string()

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

Definition at line 47 of file as.h.

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

5.6.2.24 complex__acos()

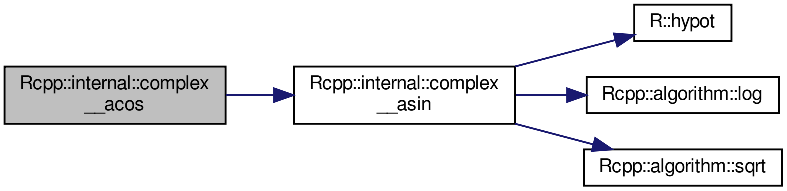
```
Rcomplex Rcpp::internal::complex__acos (
             Rcomplex z )  [inline]
```

Definition at line 148 of file complex.h.

References complex__asin().

Referenced by complex__acosh().

Here is the call graph for this function:



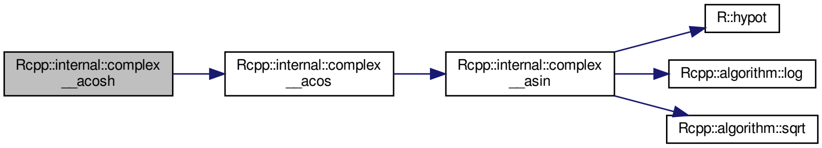
5.6.2.25 complex__acosh()

```
Rcomplex Rcpp::internal::complex__acosh (
             Rcomplex z )  [inline]
```

Definition at line 181 of file complex.h.

References complex__acos().

Here is the call graph for this function:


Rcpp::internal::complex __acosh
Rcpp::internal::complex __acos
Rcpp::internal::complex __asin
R::hypot
Rcpp::algorithm::log
Rcpp::algorithm::sqrt

5.6.2.26 complex__Arg()

```
double Rcpp::internal::complex__Arg (
             Rcomplex x )  [inline]
```

Definition at line 65 of file complex.h.

5.6.2.27 complex__asin()

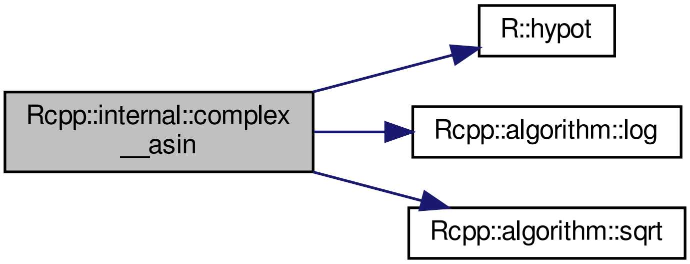
```
Rcomplex Rcpp::internal::complex__asin (
             Rcomplex z )  [inline]
```

Definition at line 132 of file complex.h.

References R::hypot(), Rcpp::algorithm::log(), and Rcpp::algorithm::sqrt().

Referenced by complex__acos(), and complex__asinh().

Here is the call graph for this function:



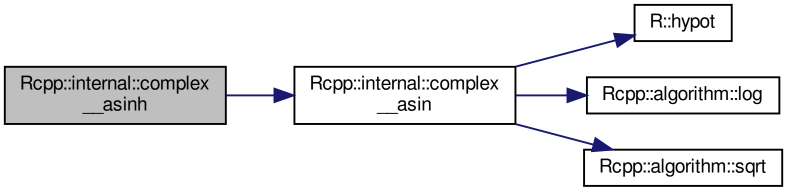
5.6.2.28 complex__asinh()

```
Rcomplex Rcpp::internal::complex__asinh (
             Rcomplex z )  [inline]
```

Definition at line 188 of file complex.h.

References complex__asin().

Here is the call graph for this function:



5.6.2.29 complex__atan()

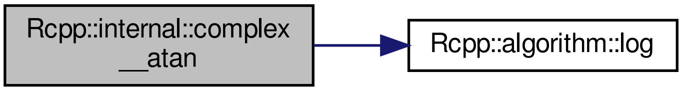
```
Rcomplex Rcpp::internal::complex__atan (
             Rcomplex z )  [inline]
```

Definition at line 164 of file complex.h.

References Rcpp::algorithm::log().

Referenced by complex__atanh().

Here is the call graph for this function:



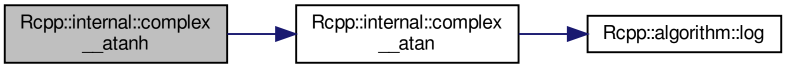
5.6.2.30 complex__atanh()

```
Rcomplex Rcpp::internal::complex__atanh (
             Rcomplex z )  [inline]
```

Definition at line 198 of file complex.h.

References complex__atan().

Here is the call graph for this function:



5.6.2.31 complex__Conj()

```
Rcomplex Rcpp::internal::complex__Conj (
             Rcomplex x )  [inline]
```

Definition at line 59 of file complex.h.

5.6.2.32 complex__cos()

```
Rcomplex Rcpp::internal::complex__cos (
             Rcomplex z )  [inline]
```

Definition at line 98 of file complex.h.

5.6.2.33 complex__cosh()

```
Rcomplex Rcpp::internal::complex__cosh (
             Rcomplex z )  [inline]
```

Definition at line 104 of file complex.h.

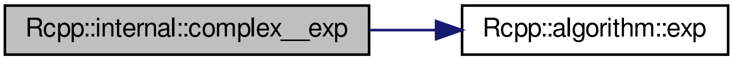
5.6.2.34 complex__exp()

```
Rcomplex Rcpp::internal::complex__exp (
             Rcomplex x )  [inline]
```

Definition at line 67 of file complex.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:



5.6.2.35 complex__Im()

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

Definition at line 57 of file complex.h.

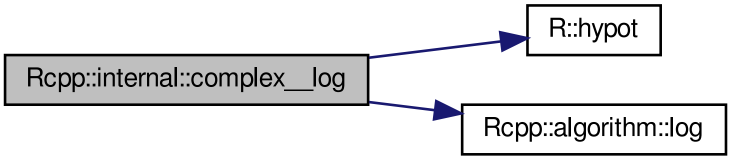
5.6.2.36 complex__log()

```
Rcomplex Rcpp::internal::complex__log (
             Rcomplex x )  [inline]
```

Definition at line 74 of file complex.h.

References R::hypot(), and Rcpp::algorithm::log().

Here is the call graph for this function:



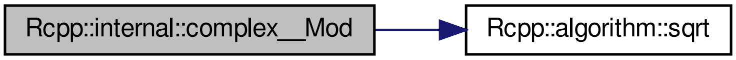
5.6.2.37 complex__Mod()

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

Definition at line 58 of file complex.h.

References Rcpp::algorithm::sqrt().

Here is the call graph for this function:



5.6.2.38 complex__Re()

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

Definition at line 56 of file complex.h.

5.6.2.39 complex__sin()

```
Rcomplex Rcpp::internal::complex__sin (
             Rcomplex z )  [inline]
```

Definition at line 110 of file complex.h.

Referenced by complex__sinh().

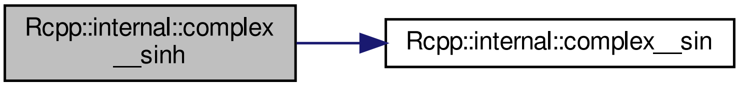
5.6.2.40 complex__sinh()

```
Rcomplex Rcpp::internal::complex__sinh (
             Rcomplex z )  [inline]
```

Definition at line 207 of file complex.h.

References complex__sin().

Here is the call graph for this function:



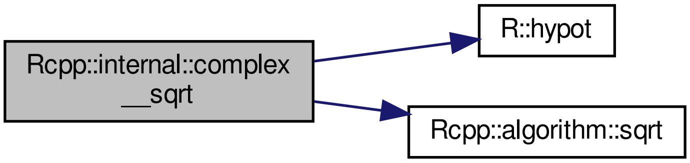
5.6.2.41 complex__sqrt()

```
Rcomplex Rcpp::internal::complex__sqrt (
             Rcomplex z )  [inline]
```

Definition at line 80 of file complex.h.

References R::hypot(), and Rcpp::algorithm::sqrt().

Here is the call graph for this function:



5.6.2.42 complex__tan()

```
Rcomplex Rcpp::internal::complex__tan (
             Rcomplex z )  [inline]
```

Definition at line 116 of file complex.h.

Referenced by complex__tanh().

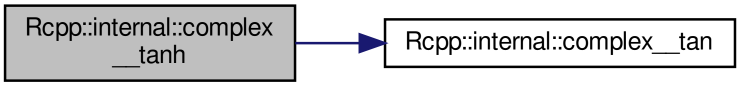
5.6.2.43 complex__tanh()

```
Rcomplex Rcpp::internal::complex__tanh (
             Rcomplex z )  [inline]
```

Definition at line 218 of file complex.h.

References complex__tan().

Here is the call graph for this function:



5.6.2.44 convert_using_rfunction()

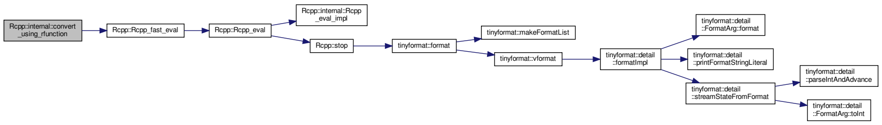
```
SEXP Rcpp::internal::convert_using_rfunction (
             SEXP x,
             const char *const fun )  [inline]
```

Definition at line 30 of file r_cast.h.

References Rcpp::Rcpp_fast_eval().

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

Here is the call graph for this function:



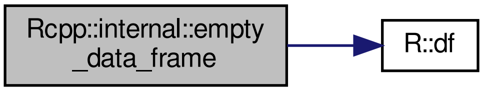
5.6.2.45 empty_data_frame()

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

Definition at line 28 of file DataFrame.h.

References R::df().

Here is the call graph for this function:



5.6.2.46 endSuspendRNGSynchronization()

```
unsigned long Rcpp::internal::endSuspendRNGSynchronization ( )  [inline]
```

Definition at line 109 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::SuspendRNGSynchronizationScope::∼SuspendRNGSynchronization↩ Scope().

5.6.2.47 enterRNGScope()

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

Definition at line 91 of file routines.h.

References GET_CALLABLE.

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

5.6.2.48 exitRNGScope()

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

Definition at line 97 of file routines.h.

References GET_CALLABLE.

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

5.6.2.49 export_range__dispatch()

```
template<typename InputIterator , typename value_type >
void Rcpp::internal::export_range__dispatch (
             SEXP x,
             InputIterator first,
             ::Rcpp::traits::r_type_generic_tag  )
```

Definition at line 29 of file export.h.

5.6.2.50 factorial()

```
double Rcpp::internal::factorial (
             double x )  [inline]
```

Definition at line 55 of file math.h.

5.6.2.51 get_column() [1/2]

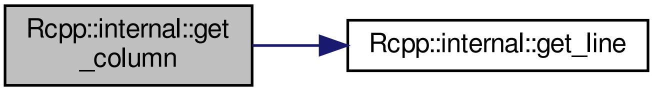
```
int Rcpp::internal::get_column (
             int index,
             int nr )  [inline]
```

Definition at line 32 of file tools.h.

References get_line().

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

Here is the call graph for this function:



5.6.2.52 get_column() [2/2]

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

Definition at line 37 of file tools.h.

5.6.2.53 get_converter_name()

```
template<typename FROM , typename TO >
std::string Rcpp::internal::get_converter_name (
            const char * from,
            const char * to )
```

Definition at line 32 of file Module.h.

5.6.2.54 get_line()

```
int Rcpp::internal::get_line (
            int index,
            int nr )  [inline]
```

Definition at line 28 of file tools.h.

Referenced by get_column(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

5.6.2.55 get_Rcpp_namespace()

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

Definition at line 121 of file routines.h.

References GET_CALLABLE.

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

5.6.2.56 get_string_buffer()

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

Definition at line 115 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

5.6.2.57 getLongjumpToken()

```
SEXP Rcpp::internal::getLongjumpToken (
             SEXP sentinel )  [inline]
```

Definition at line 144 of file exceptions.h.

Referenced by Rcpp::LongjumpException::LongjumpException(), and resumeJump().

5.6.2.58 getPosixClasses()

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

Definition at line 179 of file Date.h.

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

5.6.2.59 grow__dispatch() [1/2]

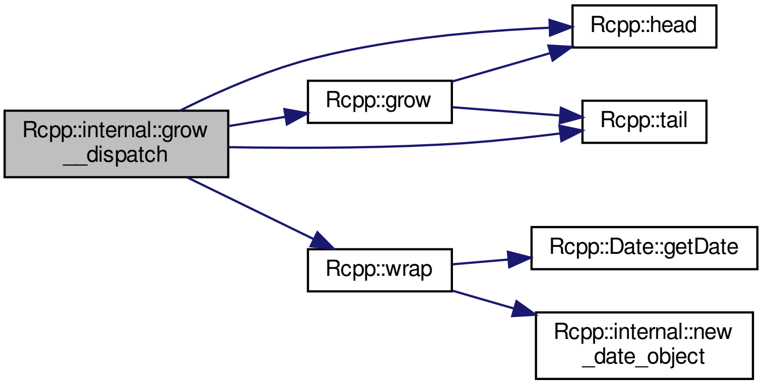
```
template<typename T >
SEXP Rcpp::internal::grow__dispatch (
             ::Rcpp::traits::false_type ,
             const T & head,
             SEXP tail )  [inline]
```

Definition at line 43 of file grow.h.

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

Referenced by Rcpp::grow().

Here is the call graph for this function:



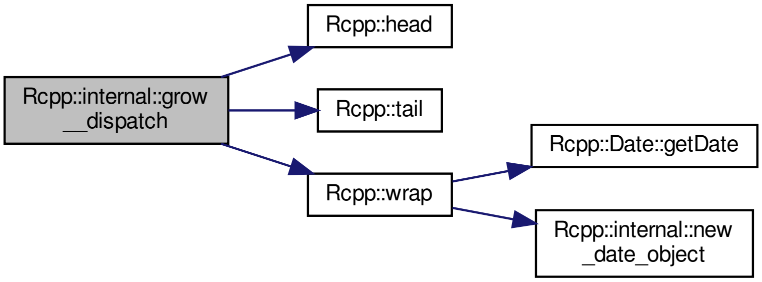
5.6.2.60 grow__dispatch() [2/2]

```
template<typename T >
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::head(), Rcpp::tail(), and Rcpp::wrap().

Here is the call graph for this function:



5.6.2.61 interruptedError()

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

Definition at line 40 of file Interrupt.h.

5.6.2.62 is__dispatch() [1/2]

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

Definition at line 38 of file is.h.

5.6.2.63 is__dispatch() [2/2]

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

Definition at line 43 of file is.h.

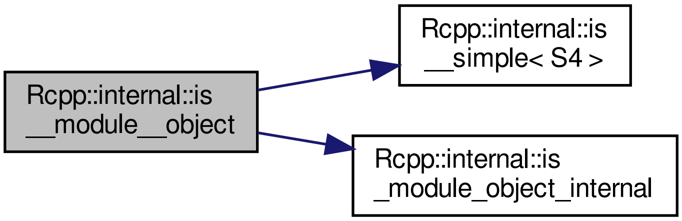
5.6.2.64 is__module__object()

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

Definition at line 160 of file is.h.

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

Here is the call graph for this function:



5.6.2.65 is__simple()

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

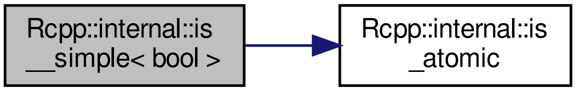
5.6.2.66 is__simple< bool >()

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

Definition at line 39 of file is.h.

References is_atomic().

Here is the call graph for this function:



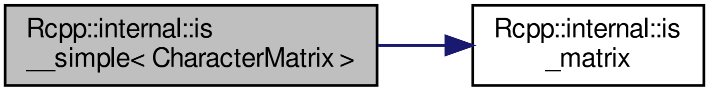
5.6.2.67 is__simple< CharacterMatrix >()

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

Definition at line 54 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.68 is__simple< CharacterVector >()

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

Definition at line 51 of file is.h.

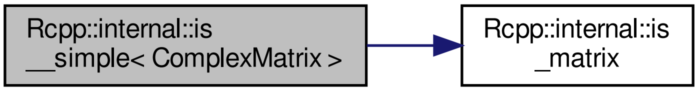
5.6.2.69 is__simple< ComplexMatrix >()

```
template<>
bool Rcpp::internal::is__simple< ComplexMatrix > (
             SEXP x )  [inline]
```

Definition at line 87 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.70 is__simple< ComplexVector >()

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

Definition at line 63 of file is.h.

5.6.2.71 is__simple< DataFrame >()

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

Definition at line 104 of file is.h.

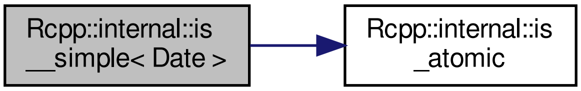
5.6.2.72 is__simple< Date >()

```
template<>
bool Rcpp::internal::is__simple< Date > (
             SEXP x )  [inline]
```

Definition at line 138 of file is.h.

References is_atomic().

Here is the call graph for this function:



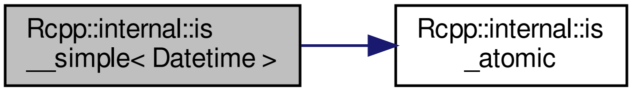
5.6.2.73 is__simple< Datetime >()

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

Definition at line 141 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.74 is__simple< DatetimeVector >()

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

Definition at line 147 of file is.h.

5.6.2.75 is__simple< DateVector >()

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

Definition at line 144 of file is.h.

5.6.2.76 is__simple< DottedPair >()

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

Definition at line 78 of file is.h.

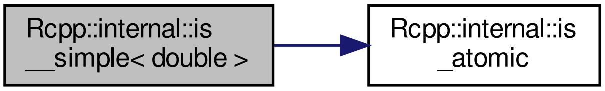
5.6.2.77 is__simple< double >()

```
template<>
bool Rcpp::internal::is__simple< double > (
             SEXP x )  [inline]
```

Definition at line 36 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.78 is__simple< Environment >()

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

Definition at line 130 of file is.h.

5.6.2.79 is__simple< Formula >()

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

Definition at line 133 of file is.h.

5.6.2.80 is__simple< Function >()

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

Definition at line 127 of file is.h.

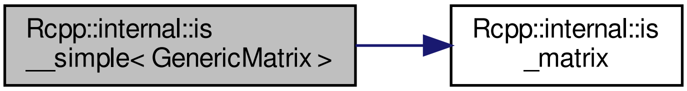
5.6.2.81 is__simple< GenericMatrix >()

```
template<>
bool Rcpp::internal::is__simple< GenericMatrix > (
            SEXP x )  [inline]
```

Definition at line 99 of file is.h.

References is_matrix().

Here is the call graph for this function:



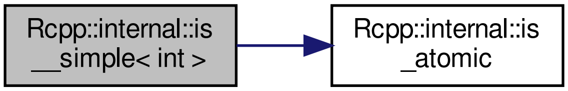
5.6.2.82 is__simple< int >()

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

Definition at line 33 of file is.h.

References is_atomic().

Here is the call graph for this function:



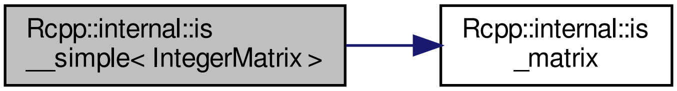
5.6.2.83 is__simple< IntegerMatrix >()

```
template<>
bool Rcpp::internal::is__simple< IntegerMatrix > (
            SEXP x )  [inline]
```

Definition at line 84 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.84 is__simple< IntegerVector >()

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

Definition at line 60 of file is.h.

5.6.2.85 is__simple< Language >()

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

Definition at line 75 of file is.h.

5.6.2.86 is__simple< List >()

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

Definition at line 81 of file is.h.

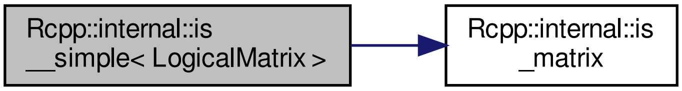
5.6.2.87 is__simple< LogicalMatrix >()

```
template<>
bool Rcpp::internal::is__simple< LogicalMatrix > (
              SEXP x )  [inline]
```

Definition at line 96 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.88 is__simple< LogicalVector >()

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

Definition at line 72 of file is.h.

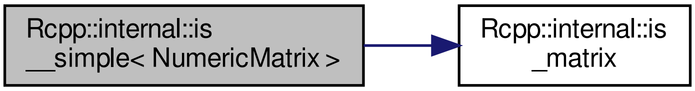
5.6.2.89 is__simple< NumericMatrix >()

```
template<>
bool Rcpp::internal::is__simple< NumericMatrix > (
             SEXP x )  [inline]
```

Definition at line 93 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.90 is__simple< NumericVector >()

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

Definition at line 69 of file is.h.

5.6.2.91 is__simple< Pairlist >()

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

Definition at line 124 of file is.h.

5.6.2.92 is__simple< Promise >()

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

Definition at line 121 of file is.h.

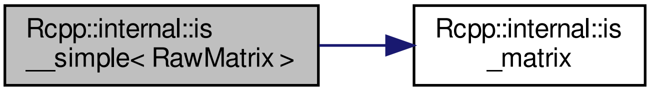
5.6.2.93 is__simple< RawMatrix >()

```
template<>
bool Rcpp::internal::is__simple< RawMatrix > (
              SEXP x )  [inline]
```

Definition at line 90 of file is.h.

References is_matrix().

Here is the call graph for this function:



5.6.2.94 is__simple< RawVector >()

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

Definition at line 66 of file is.h.

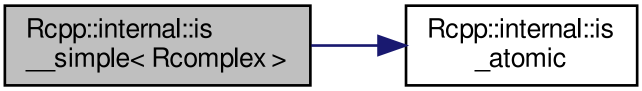
5.6.2.95 is__simple< Rcomplex >()

```
template<>
bool Rcpp::internal::is__simple< Rcomplex > (
             SEXP x )  [inline]
```

Definition at line 48 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.96 is__simple< Reference >()

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

Definition at line 117 of file is.h.

5.6.2.97 is__simple< RObject >()

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

Definition at line 57 of file is.h.

5.6.2.98 is__simple< S4 >()

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

Definition at line 114 of file is.h.

Referenced by is__module__object().

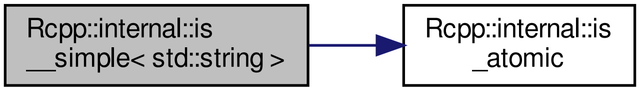
5.6.2.99 is__simple< std::string >()

```
template<>
bool Rcpp::internal::is__simple< std::string > (
             SEXP x )  [inline]
```

Definition at line 42 of file is.h.

References is_atomic().

Here is the call graph for this function:



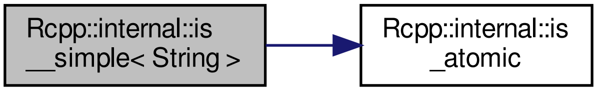
5.6.2.100 is__simple< String >()

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

Definition at line 45 of file is.h.

References is_atomic().

Here is the call graph for this function:



5.6.2.101 is__simple< Symbol >()

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

Definition at line 111 of file is.h.

5.6.2.102 is__simple< WeakReference >()

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

Definition at line 108 of file is.h.

5.6.2.103 is_atomic()

```
bool Rcpp::internal::is_atomic (
             SEXP x )  [inline]
```

Definition at line 28 of file is.h.

Referenced by is__simple< bool >(), is__simple< Date >(), is__simple< Datetime >(), is__simple< double >(), is__simple< int >(), is__simple< Rcomplex >(), is__simple< std::string >(), and is__simple< String >().

5.6.2.104 is_matrix()

```
bool Rcpp::internal::is_matrix (
             SEXP x )  [inline]
```

Definition at line 29 of file is.h.

Referenced by is__simple< CharacterMatrix >(), is__simple< ComplexMatrix >(), is__simple< GenericMatrix >(), is__simple< IntegerMatrix >(), is__simple< LogicalMatrix >(), is__simple< NumericMatrix >(), and is__simple< RawMatrix >().

5.6.2.105 is_module_object_internal()

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

Definition at line 153 of file is.h.

Referenced by is__module__object().

5.6.2.106 is_Rcpp_eval_call()

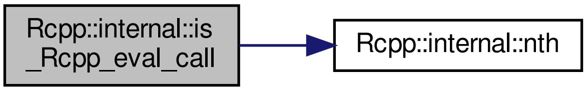
```
bool Rcpp::internal::is_Rcpp_eval_call (
             SEXP expr )  [inline]
```

Definition at line 248 of file exceptions.h.

References nth().

Referenced by get_last_call().

Here is the call graph for this function:



5.6.2.107 isLongjumpSentinel()

```
bool Rcpp::internal::isLongjumpSentinel (
             SEXP x )  [inline]
```

Definition at line 137 of file exceptions.h.

Referenced by Rcpp::LongjumpException::LongjumpException(), and resumeJump().

5.6.2.108 lfactorial()

```
double Rcpp::internal::lfactorial (
             double x )  [inline]
```

Definition at line 56 of file math.h.

5.6.2.109 longjumpSentinel()

```
SEXP Rcpp::internal::longjumpSentinel (
             SEXP token )  [inline]
```

Definition at line 126 of file exceptions.h.

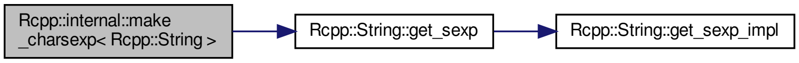
5.6.2.110 make_charsexp< Rcpp::String >()

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

Definition at line 694 of file String.h.

References Rcpp::String::get_sexp().

Here is the call graph for this function:



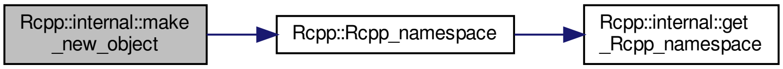
5.6.2.111 make_new_object()

```
template<typename Class >
SEXP Rcpp::internal::make_new_object (
             Class * ptr )
```

Definition at line 77 of file Module.h.

References Rcpp::Rcpp_namespace().

Here is the call graph for this function:



5.6.2.112 maybeJump()

```
void Rcpp::internal::maybeJump (
             void * unwind_data,
             Rboolean jump )  [inline]
```

Definition at line 41 of file unwindProtect.h.

References Rcpp::internal::UnwindData::jmpbuf.

Referenced by Rcpp::unwindProtect().

5.6.2.113 new_date_object()

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

Definition at line 192 of file Date.h.

Referenced by Rcpp::wrap().

5.6.2.114 new_posixt_object()

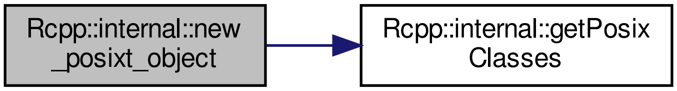
```
SEXP Rcpp::internal::new_posixt_object (
             double d )  [inline]
```

Definition at line 186 of file Date.h.

References getPosixClasses().

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

Here is the call graph for this function:



5.6.2.115 nth()

```
SEXP Rcpp::internal::nth (
             SEXP s,
             int n )  [inline]
```

Definition at line 241 of file exceptions.h.

Referenced by is_Rcpp_eval_call().

5.6.2.116 operator+() [1/2]

```
template<template< class > class StoragePolicy>
std::string Rcpp::internal::operator+ (
             const std::string & x,
             const const_string_proxy< STRSXP, StoragePolicy > & y )  [inline]
```

Definition at line 168 of file const_string_proxy.h.

5.6.2.117 operator+() [2/2]

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

Definition at line 259 of file string_proxy.h.

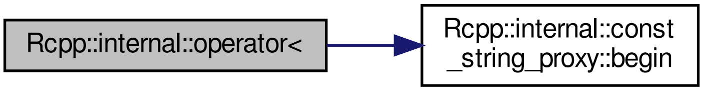
5.6.2.118 operator<() [1/2]

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

Definition at line 128 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



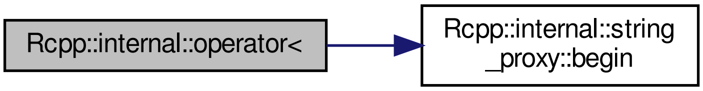
5.6.2.119 operator<() [2/2]

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

Definition at line 221 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:


Rcpp::internal::operator<
Rcpp::internal::string_proxy::begin

5.6.2.120 operator<<() [1/2]

```
template<template< class > class StoragePolicy>
std::ostream& Rcpp::internal::operator<< (
            std::ostream & os,
            const const_string_proxy< STRSXP, StoragePolicy > & proxy ) [inline]
```

Definition at line 162 of file const_string_proxy.h.

5.6.2.121 operator<<() [2/2]

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

Definition at line 254 of file string_proxy.h.

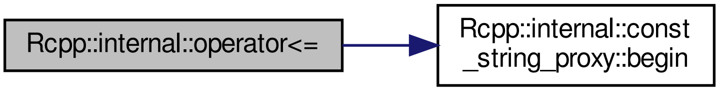
5.6.2.122 operator<=() [1/2]

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

Definition at line 152 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



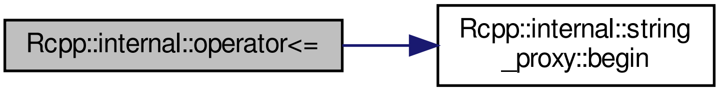
5.6.2.123 operator<=() [2/2]

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

Definition at line 245 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



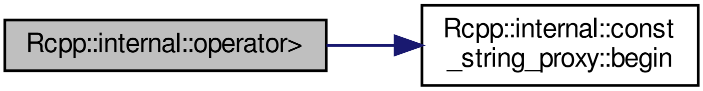
5.6.2.124 operator>() [1/2]

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

Definition at line 136 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



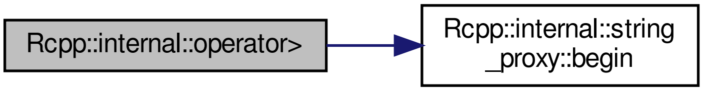
5.6.2.125 operator>() [2/2]

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

Definition at line 229 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



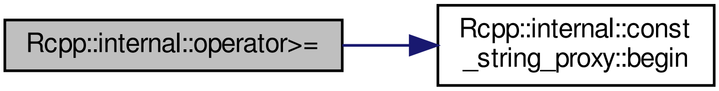
5.6.2.126 operator>=() [1/2]

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

Definition at line 144 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



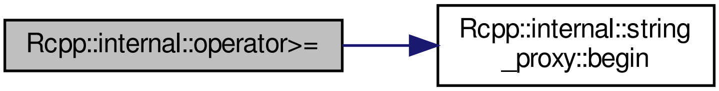
5.6.2.127 operator>=() [2/2]

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

Definition at line 237 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



5.6.2.128 primitive_as()

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

Definition at line 31 of file as.h.

5.6.2.129 r_true_cast()

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

Definition at line 46 of file r_cast.h.

5.6.2.130 r_true_cast< CPLXSXP >()

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

Definition at line 98 of file r_cast.h.

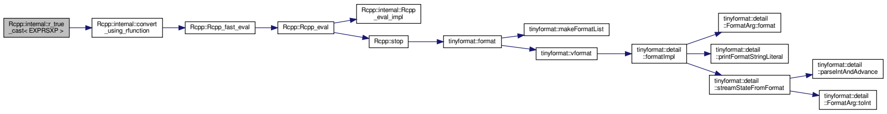
5.6.2.131 r_true_cast< EXPRSXP >()

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

Definition at line 141 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:



5.6.2.132 r_true_cast< INTSXP >()

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

Definition at line 86 of file r_cast.h.

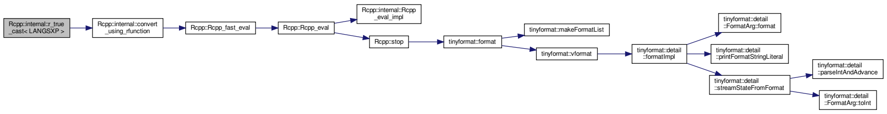
5.6.2.133 r_true_cast< LANGSXP >()

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

Definition at line 158 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:



5.6.2.134 r_true_cast< LGLSXP >()

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

Definition at line 102 of file r_cast.h.

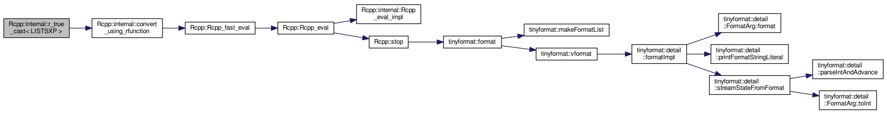
5.6.2.135 r_true_cast< LISTSXP >()

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

Definition at line 145 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:



5.6.2.136 r_true_cast< RAWSXP >()

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

Definition at line 94 of file r_cast.h.

5.6.2.137 r_true_cast< REALSXP >()

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

Definition at line 90 of file r_cast.h.

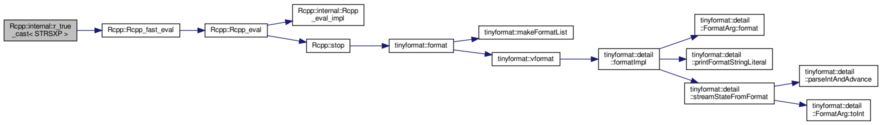
5.6.2.138 r_true_cast< STRSXP >()

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

Definition at line 107 of file r_cast.h.

References Rcpp::Rcpp_fast_eval().

Here is the call graph for this function:



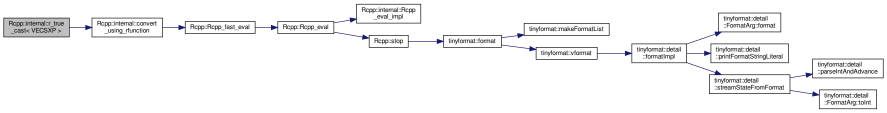
5.6.2.139 r_true_cast< VECSXP >()

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

Definition at line 137 of file r_cast.h.

References convert_using_rfunction().

Here is the call graph for this function:



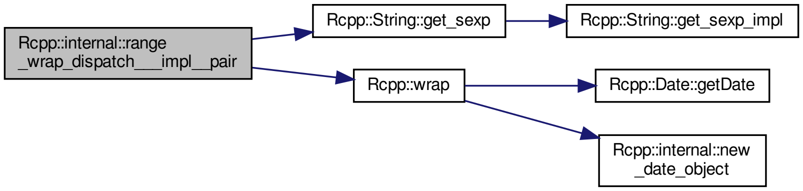
5.6.2.140 range_wrap_dispatch___impl__pair() [1/2]

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

Definition at line 47 of file wrap.h.

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

Here is the call graph for this function:



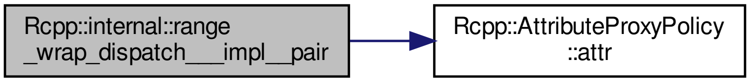
5.6.2.141 range_wrap_dispatch___impl__pair() [2/2]

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

Definition at line 29 of file wrap.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), DEMANGLE, and RCPP_DEBUG_3.

Here is the call graph for this function:



5.6.2.142 Rcpp_eval_impl()

```
SEXP Rcpp::internal::Rcpp_eval_impl (
            SEXP expr,
            SEXP env )  [inline]
```

Definition at line 48 of file Rcpp_eval.h.

Referenced by Rcpp::fast_eval(), Rcpp::find(), Rcpp::get(), and Rcpp::Rcpp_eval().

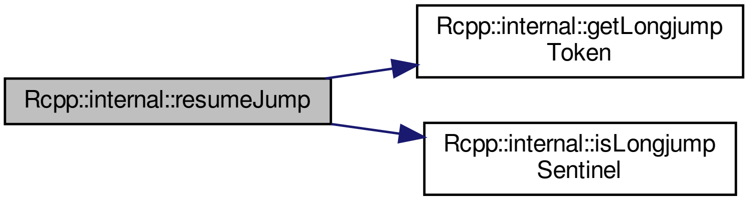
5.6.2.143 resumeJump()

```
void Rcpp::internal::resumeJump (
            SEXP token )  [inline]
```

Definition at line 148 of file exceptions.h.

References getLongjumpToken(), and isLongjumpSentinel().

Here is the call graph for this function:


Rcpp::internal::resumeJump
Rcpp::internal::getLongjump Token
Rcpp::internal::isLongjump Sentinel

5.6.2.144 vector_from_string()

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

Definition at line 29 of file vector_from_string.h.

5.6.2.145 vector_from_string< EXPRSXP >()

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

Definition at line 50 of file vector_from_string.h.

5.6.2.146 vector_from_string_expr()

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

Definition at line 35 of file vector_from_string.h.

5.6.2.147 wrap_range_sugar_expression()

```
template<typename T >
SEXP Rcpp::internal::wrap_range_sugar_expression (
             const T & object,
             Rcpp::traits::true_type  )  [inline]
```

Definition at line 30 of file Vector.h.

References DEMANGLE, and RCPP_DEBUG_1.

5.6.3 Variable Documentation

5.6.3.1 rngSynchronizationSuspended

```
int Rcpp::internal::rngSynchronizationSuspended = 0
```

Definition at line 53 of file api.cpp.

5.7 Rcpp::internal::debug Namespace Reference

Functions

- std::string short_file_name (const char ∗file)

5.7.1 Function Documentation

5.7.1.1 short_file_name()

```
std::string Rcpp::internal::debug::short_file_name (
              const char * file )  [inline]
```

Definition at line 31 of file macros.h.

Referenced by registerFunctions().

5.8 Rcpp::InternalFunctionWithStdFunction Namespace Reference

Classes

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

5.9 Rcpp::stats Namespace Reference

Classes

- class D0
- class D1
- class D2
- class D3
- class P0
- class P1
- class P2
- class P3
- class Q0
- class Q1
- class Q2
- class Q3
- class BetaGenerator
- class BinomGenerator
- class CauchyGenerator
- class CauchyGenerator_1
- class CauchyGenerator_0
- class ChisqGenerator
- class ExpGenerator
- class ExpGenerator__rate1
- class FGenerator_Finite_Finite

- class FGenerator_NotFinite_Finite
- class FGenerator_Finite_NotFinite
- class GammaGenerator
- class GeomGenerator
- class HyperGenerator
- class LNormGenerator
- class LNormGenerator_1
- class LNormGenerator_0
- class LogisGenerator
- class LogisGenerator_1
- class LogisGenerator_0
- class NBinomGenerator
- class NBinomGenerator_Mu
- class NChisqGenerator
- class NormGenerator
- class NormGenerator__sd1
- class NormGenerator__mean0
- class NormGenerator__mean0__sd1
- class PoissonGenerator
- class SignRankGenerator
- class TGenerator
- class UnifGenerator
- class UnifGenerator__0__1
- class WeibullGenerator
- class WeibullGenerator__scale1
- class WilcoxGenerator

Functions

- double dcauchy_0 (double x, int give_log)
- double dcauchy_1 (double x, double location, int give_log)
- double pcauchy_0 (double x, int lower_tail, int log_p)
- double pcauchy_1 (double x, double location, int lower_tail, int log_p)
- double qcauchy_0 (double p, int lower_tail, int log_p)
- double qcauchy_1 (double p, double location, int lower_tail, int log_p)
- double d_exp_0 (double x, int give_log)
- double q_exp_0 (double p, int lower_tail, int log_p)
- double p_exp_0 (double x, int lower_tail, int log_p)
- double dgamma_1 (double x, double shape, int log_p)
- double pgamma_1 (double x, double 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)

5.9.1 Function Documentation

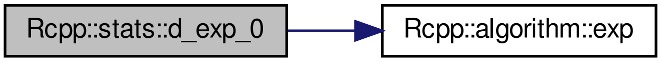
5.9.1.1 d_exp_0()

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

Definition at line 30 of file exp.h.

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

Here is the call graph for this function:



5.9.1.2 dcauchy_0()

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

Definition at line 28 of file cauchy.h.

References give_log.

5.9.1.3 dcauchy_1()

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

Definition at line 32 of file cauchy.h.

References give_log.

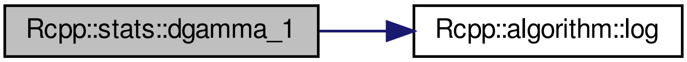
5.9.1.4 dgamma_1()

```
double Rcpp::stats::dgamma_1 (
             double x,
             double shape,
             int log_p )  [inline]
```

Definition at line 30 of file gamma.h.

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

Here is the call graph for this function:



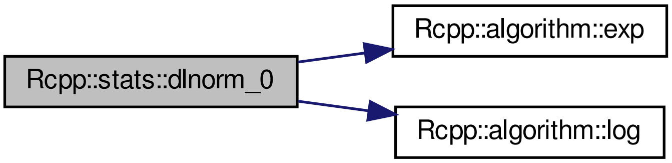
5.9.1.5 dlnorm_0()

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

Definition at line 30 of file lnorm.h.

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

Here is the call graph for this function:



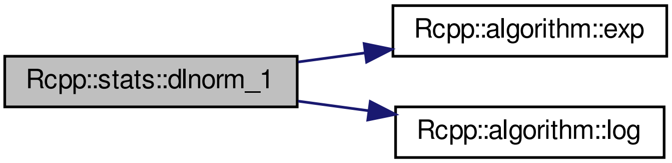
5.9.1.6 dlnorm_1()

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

Definition at line 47 of file lnorm.h.

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

Here is the call graph for this function:



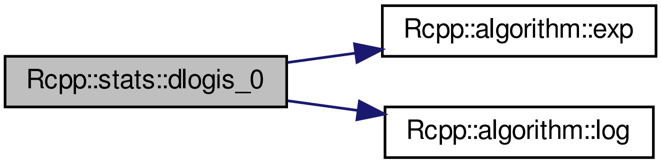
5.9.1.7 dlogis_0()

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

Definition at line 28 of file logis.h.

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

Here is the call graph for this function:



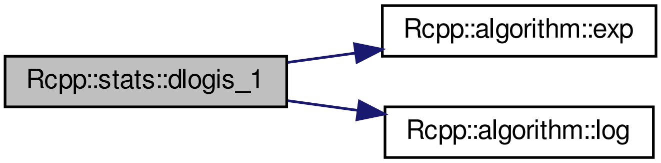
5.9.1.8 dlogis_1()

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

Definition at line 40 of file logis.h.

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

Here is the call graph for this function:



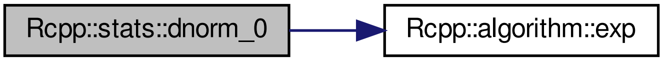
5.9.1.9 dnorm_0()

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

Definition at line 43 of file norm.h.

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

Here is the call graph for this function:



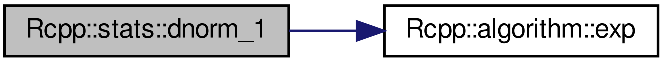
5.9.1.10 dnorm_1()

```
double Rcpp::stats::dnorm_1 (
             double x,
             double mu,
             int give_log )  [inline]
```

Definition at line 28 of file norm.h.

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

Here is the call graph for this function:



5.9.1.11 dunif_0()

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

Definition at line 34 of file unif.h.

References give_log, and R_D__0.

5.9.1.12 dunif_1()

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

Definition at line 30 of file unif.h.

References give_log.

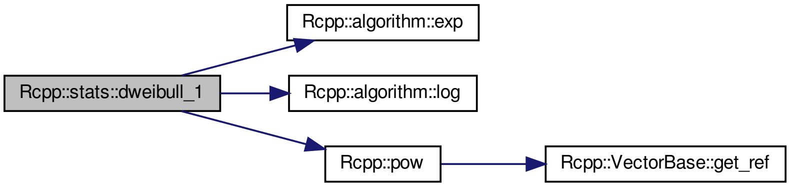
5.9.1.13 dweibull_1()

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

Definition at line 30 of file weibull.h.

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

Here is the call graph for this function:



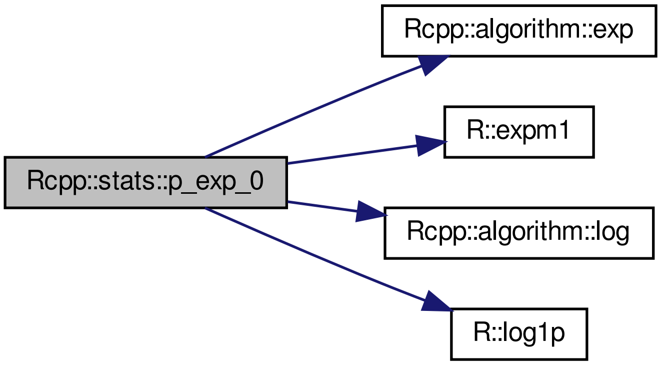
5.9.1.14 p_exp_0()

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

Definition at line 54 of file exp.h.

References Rcpp::algorithm::exp(), R::expm1(), Rcpp::algorithm::log(), R::log1p(), R_D_exp, and R_DT_0.

Here is the call graph for this function:



5.9.1.15 pcauchy_0()

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

Definition at line 36 of file cauchy.h.

5.9.1.16 pcauchy_1()

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

Definition at line 40 of file cauchy.h.

5.9.1.17 pgamma_1()

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

Definition at line 56 of file gamma.h.

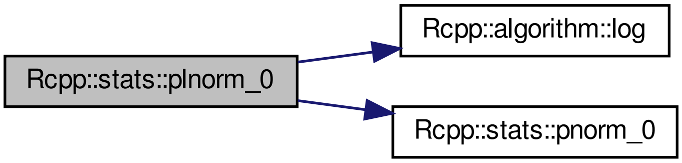
5.9.1.18 plnorm_0()

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

Definition at line 65 of file lnorm.h.

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

Here is the call graph for this function:



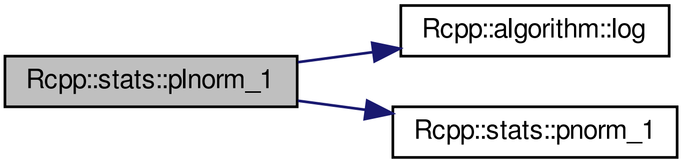
5.9.1.19 plnorm_1()

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

Definition at line 76 of file lnorm.h.

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

Here is the call graph for this function:



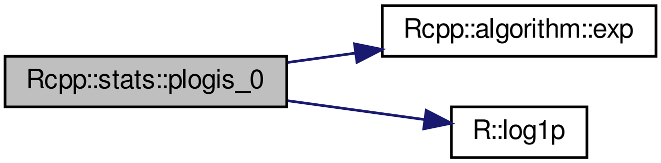
5.9.1.20 plogis_0()

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

Definition at line 54 of file logis.h.

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

Here is the call graph for this function:



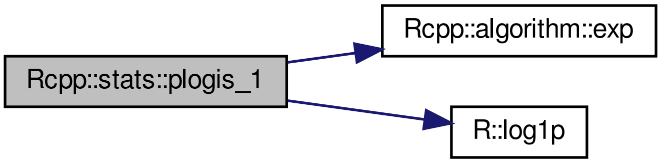
5.9.1.21 plogis_1()

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

Definition at line 69 of file logis.h.

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

Here is the call graph for this function:



5.9.1.22 pnorm_0()

```
double Rcpp::stats::pnorm_0 (
             double x,
             int lower_tail,
             int log_p )  [inline]
```

Definition at line 78 of file norm.h.

References R_DT_0, and R_DT_1.

Referenced by plnorm_0().

5.9.1.23 pnorm_1()

```
double Rcpp::stats::pnorm_1 (
             double x,
             double mu,
             int lower_tail,
             int log_p )  [inline]
```

Definition at line 56 of file norm.h.

References ML_NAN, R_DT_0, and R_DT_1.

Referenced by plnorm_1().

5.9.1.24 punif_0()

```
double Rcpp::stats::punif_0 (
             double x,
             int lower_tail,
             int log_p )  [inline]
```

Definition at line 50 of file unif.h.

References R_D_val, R_DT_0, and R_DT_1.

5.9.1.25 punif_1()

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

Definition at line 46 of file unif.h.

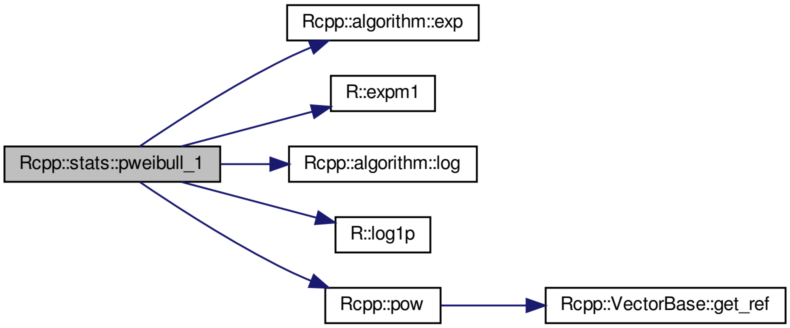
5.9.1.26 pweibull_1()

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

Definition at line 49 of file weibull.h.

References Rcpp::algorithm::exp(), R::expm1(), Rcpp::algorithm::log(), R::log1p(), Rcpp::pow(), R_D_exp, and R_DT↩ _0.

Here is the call graph for this function:



5.9.1.27 q_exp_0()

```
double Rcpp::stats::q_exp_0 (
            double p,
            int lower_tail,
            int log_p )  [inline]
```

Definition at line 42 of file exp.h.

References R_DT_0, and R_DT_Clog.

5.9.1.28 qcauchy_0()

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

Definition at line 44 of file cauchy.h.

5.9.1.29 qcauchy_1()

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

Definition at line 47 of file cauchy.h.

5.9.1.30 qgamma_1()

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

Definition at line 59 of file gamma.h.

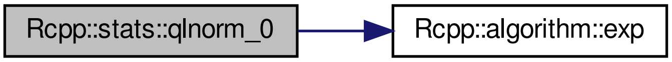
5.9.1.31 qlnorm_0()

```
double Rcpp::stats::qlnorm_0 (
             double p,
             int lower_tail,
             int log_p )  [inline]
```

Definition at line 87 of file lnorm.h.

References Rcpp::algorithm::exp(), ML_POSINF, and R_Q_P01_boundaries.

Here is the call graph for this function:



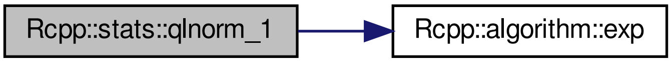
5.9.1.32 qlnorm_1()

```
double Rcpp::stats::qlnorm_1 (
             double p,
             double meanlog,
             int lower_tail,
             int log_p )  [inline]
```

Definition at line 97 of file lnorm.h.

References Rcpp::algorithm::exp(), ML_POSINF, and R_Q_P01_boundaries.

Here is the call graph for this function:



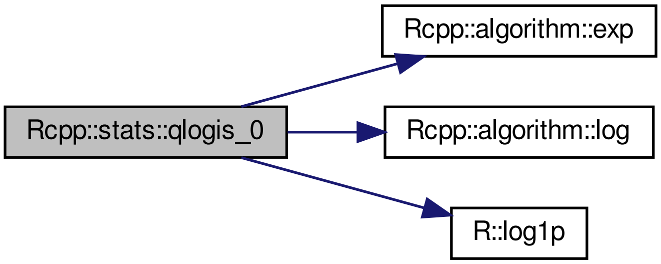
5.9.1.33 qlogis_0()

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

Definition at line 84 of file logis.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), R::log1p(), ML_NEGINF, ML_POSINF, and R_Q_P01_↩ boundaries.

Here is the call graph for this function:



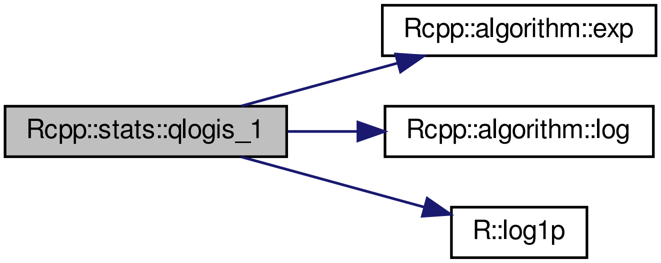
5.9.1.34 qlogis_1()

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

Definition at line 106 of file logis.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), R::log1p(), ML_NEGINF, ML_POSINF, and R_Q_P01_↩ boundaries.

Here is the call graph for this function:



5.9.1.35 qnorm_0()

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

Definition at line 103 of file norm.h.

5.9.1.36 qnorm_1()

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

Definition at line 98 of file norm.h.

5.9.1.37 qunif_0()

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

Definition at line 79 of file unif.h.

References R_DT_qIv, and R_Q_P01_check.

5.9.1.38 qunif_1()

```
double Rcpp::stats::qunif_1 (
            double p,
            double a,
            int lower_tail,
            int log_p ) [inline]
```

Definition at line 65 of file unif.h.

References R_DT_qIv, and R_Q_P01_check.

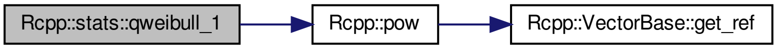
5.9.1.39 qweibull_1()

```
double Rcpp::stats::qweibull_1 (
            double p,
            double shape,
            int lower_tail,
            int log_p )  [inline]
```

Definition at line 67 of file weibull.h.

References ML_POSINF, Rcpp::pow(), R_DT_Clog, and R_Q_P01_boundaries.

Here is the call graph for this function:



5.10 Rcpp::sugar Namespace Reference

Namespaces

- cbind_impl
- detail
- median_detail

Classes

- class IndexHash
- class SelfHash
- class SugarBlock_1
- class SugarBlock_2
- class SugarBlock_2__VP
- class SugarBlock_2__PV
- class SugarBlock_3_VVV
- class SugarMath_1
- class SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >
- class SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >
- class Vectorized
- class Vectorized_INTSXP
- class Vectorized_INTSXP< Func, false, VEC >
- class All
- class All< false, T >
- class Any

- class Any< false, T >
- struct clamp_operator
- struct clamp_operator< REALSXP, true >
- class Clamp_Primitive_Vector_Primitive
- class SugarComplex
- class Cummax
- class Cummin
- class Cumprod
- class Cumsum
- class Diff
- class Diff< REALSXP, LHS_NA, LHS_T >
- class Diff< RTYPE, false, LHS_T >
- class Head
- class IfElse
- class IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class IfElse_Primitive_Vector
- class IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >
- class IfElse_Vector_Primitive
- class IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >
- class IfElse_Primitive_Primitive
- class IfElse_Primitive_Primitive< RTYPE, false, COND_T >
- class IsFinite
- class IsInfinite
- class IsNa
- class IsNa< RTYPE, false, VEC_TYPE >
- class IsNa_Vector_is_na
- class IsNaN
- class Lapply
- class Lazy
- class Mapply_2
- class Mapply_2_Vector_Primitive
- class Mapply_2_Primitive_Vector
- class Mapply_3
- class Max
- class Max< RTYPE, false, T >
- class Mean
- class Mean< CPLXSXP, NA, T >
- class Mean< LGLSXP, NA, T >
- class Mean< INTSXP, NA, T >
- class Median
- class Median< RTYPE, NA, T, true >
- class Median< RTYPE, false, T, NA_RM >
- class Median< STRSXP, NA, T, NA_RM >
- class Median< STRSXP, NA, T, true >
- class Median< STRSXP, false, T, true >
- class Min
- class Min< RTYPE, false, T >
- struct pmax_op
- struct pmax_op< REALSXP, true, true >
- struct pmax_op< REALSXP, true, false >
- struct pmax_op< REALSXP, false, true >

- struct pmax_op< REALSXP, false, false >
- struct pmax_op< INTSXP, LHS_NA, RHS_NA >
- class pmax_op_Vector_Primitive
- class pmax_op_Vector_Primitive< REALSXP, true >
- class Pmax_Vector_Vector
- class Pmax_Vector_Primitive
- struct pmin_op
- struct pmin_op< REALSXP, true, true >
- struct pmin_op< REALSXP, true, false >
- struct pmin_op< REALSXP, false, true >
- struct pmin_op< REALSXP, false, false >
- struct pmin_op< INTSXP, LHS_NA, RHS_NA >
- class pmin_op_Vector_Primitive
- class pmin_op_Vector_Primitive< REALSXP, true >
- class Pmin_Vector_Vector
- class Pmin_Vector_Primitive
- class Pow
- class Pow< INTSXP, NA, T, EXPONENT_TYPE >
- class Pow< INTSXP, false, T, EXPONENT_TYPE >
- class Range
- class Range< RTYPE, false, T >
- class Rep
- class Rep_Single
- class Rep_each
- class Rep_len
- class Rev
- class RowSumsImpl
- class RowSumsImpl< RTYPE, NA, T, true >
- class RowSumsImpl< RTYPE, false, T, NA_RM >
- class ColSumsImpl
- class ColSumsImpl< RTYPE, NA, T, true >
- class ColSumsImpl< RTYPE, false, T, NA_RM >
- class RowMeansImpl
- class RowMeansImpl< RTYPE, NA, T, true >
- class RowMeansImpl< RTYPE, false, T, NA_RM >
- class ColMeansImpl
- class ColMeansImpl< RTYPE, NA, T, true >
- class ColMeansImpl< RTYPE, false, T, NA_RM >
- struct sapply_application_result_of
- class Sapply
- class Sapply< RTYPE, NA, T, Function, true >
- class Sd
- class SelfInserter
- class SelfMatch
- class SeqLen
- class RemoveFromSet
- class SetDiff
- class SetEqual
- class Intersect
- class Union
- class sign__impl

- class sign__impl< false, RTYPE >
- class Sign
- class Sum
- class Sum< REALSXP, NA, T >
- class Sum< RTYPE, false, T >
- class CountInserter
- class Grabber
- class Table
- class Tail
- class InSet
- class In
- class Var
- class Var< CPLXSXP, NA, T >
- class WhichMax
- class WhichMax< RTYPE, false, T >
- class WhichMin
- class WhichMin< RTYPE, false, T >
- class And_SingleLogicalResult_SingleLogicalResult
- class And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class And_SingleLogicalResult_bool
- class And_LogicalExpression_LogicalExpression
- class And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
- struct negate
- struct negate< false >
- class Negate_SingleLogicalResult
- class Or_SingleLogicalResult_SingleLogicalResult
- class Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class Or_SingleLogicalResult_bool
- class Or_LogicalExpression_LogicalExpression
- class Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
- class forbidden_conversion
- class forbidden_conversion< true >
- class conversion_to_bool_is_forbidden
- class SingleLogicalResult
- class Col
- class Diag_Extractor
- class Diag_Maker
- struct diag_result_type_trait
- class LowerTri
- class Outer
- class Row
- class UpperTri
- class Nona

- class Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >
- class NonaPrimitive
- class Comparator
- class Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >
- class Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >
- class Comparator_With_One_Value
- class Comparator_With_One_Value< RTYPE, Operator, false, T >
- class Divides_Vector_Vector
- class Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Divides_Vector_Primitive
- class Divides_Vector_Primitive< REALSXP, NA, T >
- class Divides_Vector_Primitive< RTYPE, false, T >
- class Divides_Vector_Primitive< REALSXP, false, T >
- class Divides_Primitive_Vector
- class Divides_Primitive_Vector< REALSXP, NA, T >
- class Divides_Primitive_Vector< RTYPE, false, T >
- class Divides_Primitive_Vector< REALSXP, false, T >
- class Minus_Vector_Vector
- class Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Minus_Vector_Primitive
- class Minus_Vector_Primitive< REALSXP, NA, T >
- class Minus_Vector_Primitive< RTYPE, false, T >
- class Minus_Vector_Primitive< REALSXP, false, T >
- class Minus_Primitive_Vector
- class Minus_Primitive_Vector< REALSXP, NA, T >
- class Minus_Primitive_Vector< RTYPE, false, T >
- class Minus_Primitive_Vector< REALSXP, false, T >
- class not_
- class not_< RTYPE, false >
- class not_< REALSXP, NA >
- class not_< REALSXP, false >
- class not_< CPLXSXP, NA >
- class not_< CPLXSXP, false >
- class Not_Vector
- class Plus_Vector_Vector
- class Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

- class Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Plus_Vector_Primitive
- class Plus_Vector_Primitive< REALSXP, NA, T >
- class Plus_Vector_Primitive< RTYPE, false, T >
- class Plus_Vector_Primitive< REALSXP, false, T >
- class Plus_Vector_Primitive_nona
- class Plus_Vector_Primitive_nona< REALSXP, NA, T >
- class Plus_Vector_Primitive_nona< RTYPE, false, T >
- class Plus_Vector_Primitive_nona< REALSXP, false, T >
- class Times_Vector_Vector
- class Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Times_Vector_Primitive
- class Times_Vector_Primitive< REALSXP, NA, T >
- class Times_Vector_Primitive< RTYPE, false, T >
- class Times_Vector_Primitive< REALSXP, false, T >
- class Times_Vector_Primitive_nona
- class Times_Vector_Primitive_nona< REALSXP, NA, T >
- class Times_Vector_Primitive_nona< RTYPE, false, T >
- class Times_Vector_Primitive_nona< REALSXP, false, T >
- struct unary_minus_result_type
- struct unary_minus_result_type< LGLSXP >
- class unary_minus
- class unary_minus< RTYPE, false >
- class unary_minus< CPLXSXP, NA >
- class unary_minus< CPLXSXP, false >
- class UnaryMinus_Vector
- class SugarIterator
- struct sugar_const_iterator_type
- struct sugar_const_iterator_type< Rcpp::Vector< RTYPE > >
- struct sugar_const_iterator_type< CharacterVector >
- struct is_sugar_vector
- struct is_sugar_vector< Rcpp::Vector< RTYPE > >

Typedefs

- typedef double(∗ DDFun) (double)
- typedef Nullable< Vector< REALSXP > > probs_t

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)
- void Normalize (Vector< REALSXP > &p, int require_k, bool replace)
- Vector< INTSXP > SampleReplace (Vector< REALSXP > &p, int n, int k, bool one_based)
- template<int RTYPE>
Vector< RTYPE > SampleReplace (Vector< REALSXP > &p, int k, const Vector< RTYPE > &ref)
- Vector< INTSXP > WalkerSample (const Vector< REALSXP > &p, int n, int nans, bool one_based)
- template<int RTYPE>
Vector< RTYPE > WalkerSample (const Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > SampleNoReplace (Vector< REALSXP > &p, int n, int nans, bool one_based)
- template<int RTYPE>
Vector< RTYPE > SampleNoReplace (Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > EmpiricalSample (int n, int size, bool replace, bool one_based)
- template<int RTYPE>
Vector< RTYPE > EmpiricalSample (int size, bool replace, const Vector< RTYPE > &ref)
- template<typename Iterator >
String collapse__impl (Iterator it, R_xlen_t n)
- template<typename T >
sugar_const_iterator_type< T >::type get_const_begin__impl (const T &obj, Rcpp::traits::true_type)
- template<typename T >
sugar_const_iterator_type< T >::type get_const_begin__impl (const T &obj, Rcpp::traits::false_type)
- template<typename T >
sugar_const_iterator_type< T >::type get_const_begin (const T &obj)
- template<> SEXP ∗ get_const_begin (const CharacterVector &obj)
- template<typename T >
sugar_const_iterator_type< T >::type get_const_end (const T &obj)

5.10.1 Typedef Documentation

5.10.1.1 DDFun

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

Definition at line 28 of file Vectorized_Math.h.

5.10.1.2 probs_t

```
typedef Nullable< Vector<REALSXP> > Rcpp::sugar::probs_t
```

Definition at line 395 of file sample.h.

5.10.2 Function Documentation

5.10.2.1 cbind() [1/49]

```
template<typename T1 , typename T2 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2 )  [inline]
```

Definition at line 516 of file cbind.h.

References MakeBindable.

5.10.2.2 cbind() [2/49]

```
template<typename T1 , typename T2 , typename T3 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3 )  [inline]
```

Definition at line 522 of file cbind.h.

References MakeBindable.

5.10.2.3 cbind() [3/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4 )  [inline]
```

Definition at line 528 of file cbind.h.

References MakeBindable.

5.10.2.4 cbind() [4/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5 )  [inline]
```

Definition at line 535 of file cbind.h.

References MakeBindable.

5.10.2.5 cbind() [5/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6 )  [inline]
```

Definition at line 541 of file cbind.h.

References MakeBindable.

5.10.2.6 cbind() [6/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
                const T1 & t1,
                const T2 & t2,
                const T3 & t3,
                const T4 & t4,
                const T5 & t5,
                const T6 & t6,
                const T7 & t7 ) [inline]
```

Definition at line 547 of file cbind.h.

References MakeBindable.

5.10.2.7 cbind() [7/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
                const T1 & t1,
                const T2 & t2,
                const T3 & t3,
                const T4 & t4,
                const T5 & t5,
                const T6 & t6,
                const T7 & t7,
                const T8 & t8 ) [inline]
```

Definition at line 553 of file cbind.h.

References MakeBindable.

5.10.2.8 cbind() [8/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
                const T1 & t1,
                const T2 & t2,
                const T3 & t3,
                const T4 & t4,
                const T5 & t5,
                const T6 & t6,
                const T7 & t7,
                const T8 & t8,
                const T9 & t9 ) [inline]
```

Definition at line 559 of file cbind.h.

References MakeBindable.

5.10.2.9 cbind() [9/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10 )  [inline]
```

Definition at line 566 of file cbind.h.

References MakeBindable.

5.10.2.10 cbind() [10/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11 )  [inline]
```

Definition at line 572 of file cbind.h.

References MakeBindable.

5.10.2.11 cbind() [11/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12 )  [inline]
```

Definition at line 578 of file cbind.h.

References MakeBindable.

5.10.2.12 cbind() [12/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13 )  [inline]
```

Definition at line 584 of file cbind.h.

References MakeBindable.

5.10.2.13 cbind() [13/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14 )  [inline]
```

Definition at line 590 of file cbind.h.

References MakeBindable.

5.10.2.14 cbind() [14/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15 )  [inline]
```

Definition at line 597 of file cbind.h.

References MakeBindable.

5.10.2.15 cbind() [15/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16 )  [inline]
```

Definition at line 603 of file cbind.h.

References MakeBindable.

5.10.2.16 cbind() [16/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17 )  [inline]
```

Definition at line 609 of file cbind.h.

References MakeBindable.

5.10.2.17 cbind() [17/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18 )  [inline]
```

Definition at line 615 of file cbind.h.

References MakeBindable.

5.10.2.18 cbind() [18/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19 )  [inline]
```

Definition at line 621 of file cbind.h.

References MakeBindable.

5.10.2.19 cbind() [19/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20 )  [inline]
```

Definition at line 628 of file cbind.h.

References MakeBindable.

5.10.2.20 cbind() [20/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21 )  [inline]
```

Definition at line 634 of file cbind.h.

References MakeBindable.

5.10.2.21 cbind() [21/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22 )  [inline]
```

Definition at line 640 of file cbind.h.

References MakeBindable.

5.10.2.22 cbind() [22/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23 )  [inline]
```

Definition at line 646 of file cbind.h.

References MakeBindable.

5.10.2.23 cbind() [23/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24 )  [inline]
```

Definition at line 652 of file cbind.h.

References MakeBindable.

5.10.2.24 cbind() [24/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25 )  [inline]
```

Definition at line 659 of file cbind.h.

References MakeBindable.

5.10.2.25 cbind() [25/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26 )  [inline]
```

Definition at line 665 of file cbind.h.

References MakeBindable.

5.10.2.26 cbind() [26/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27 )  [inline]
```

Definition at line 671 of file cbind.h.

References MakeBindable.

5.10.2.27 cbind() [27/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28 )  [inline]
```

Definition at line 677 of file cbind.h.

References MakeBindable.

5.10.2.28 cbind() [28/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29 )  [inline]
```

Definition at line 683 of file cbind.h.

References MakeBindable.

5.10.2.29 cbind() [29/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30 )  [inline]
```

Definition at line 690 of file cbind.h.

References MakeBindable.

5.10.2.30 cbind() [30/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31 )  [inline]
```

Definition at line 696 of file cbind.h.

References MakeBindable.

5.10.2.31 cbind() [31/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32 )  [inline]
```

Definition at line 702 of file cbind.h.

References MakeBindable.

5.10.2.32 cbind() [32/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33 )  [inline]
```

Definition at line 708 of file cbind.h.

References MakeBindable.

5.10.2.33 cbind() [33/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34 )  [inline]
```

Definition at line 714 of file cbind.h.

References MakeBindable.

5.10.2.34 cbind() [34/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35 )  [inline]
```

Definition at line 721 of file cbind.h.

References MakeBindable.

5.10.2.35 cbind() [35/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36 )  [inline]
```

Definition at line 727 of file cbind.h.

References MakeBindable.

5.10.2.36 cbind() [36/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33,
            const T34 & t34,
            const T35 & t35,
            const T36 & t36,
            const T37 & t37 )  [inline]
```

Definition at line 733 of file cbind.h.

References MakeBindable.

5.10.2.37 cbind() [37/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38 )  [inline]
```

Definition at line 739 of file cbind.h.

References MakeBindable.

5.10.2.38 cbind() [38/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33,
            const T34 & t34,
            const T35 & t35,
            const T36 & t36,
            const T37 & t37,
            const T38 & t38,
            const T39 & t39 )  [inline]
```

Definition at line 745 of file cbind.h.

References MakeBindable.

5.10.2.39 cbind() [39/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40 )  [inline]
```

Definition at line 752 of file cbind.h.

References MakeBindable.

5.10.2.40 cbind() [40/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41 )  [inline]
```

Definition at line 758 of file cbind.h.

References MakeBindable.

5.10.2.41 cbind() [41/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33,
            const T34 & t34,
            const T35 & t35,
            const T36 & t36,
            const T37 & t37,
            const T38 & t38,
            const T39 & t39,
            const T40 & t40,
            const T41 & t41,
            const T42 & t42 )  [inline]
```

Definition at line 764 of file cbind.h.

References MakeBindable.

5.10.2.42 cbind() [42/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43 )  [inline]
```

Definition at line 770 of file cbind.h.

References MakeBindable.

5.10.2.43 cbind() [43/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
```

```
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43,
             const T44 & t44 )  [inline]
```

Definition at line 776 of file cbind.h.

References MakeBindable.

5.10.2.44 cbind() [44/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43,
             const T44 & t44,
             const T45 & t45 )  [inline]
```

Definition at line 783 of file cbind.h.

References MakeBindable.

5.10.2.45 cbind() [45/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43,
             const T44 & t44,
             const T45 & t45,
             const T46 & t46 )  [inline]
```

Definition at line 789 of file cbind.h.

References MakeBindable.

5.10.2.46 cbind() [46/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33,
            const T34 & t34,
            const T35 & t35,
            const T36 & t36,
            const T37 & t37,
            const T38 & t38,
            const T39 & t39,
            const T40 & t40,
            const T41 & t41,
            const T42 & t42,
            const T43 & t43,
            const T44 & t44,
            const T45 & t45,
            const T46 & t46,
            const T47 & t47 )  [inline]
```

Definition at line 795 of file cbind.h.

References MakeBindable.

5.10.2.47 cbind() [47/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 , typename T48 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43,
             const T44 & t44,
             const T45 & t45,
             const T46 & t46,
             const T47 & t47,
             const T48 & t48 )  [inline]
```

Definition at line 801 of file cbind.h.

References MakeBindable.

5.10.2.48 cbind() [48/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
```

```
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 , typename T48 , typename T49 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
             const T1 & t1,
             const T2 & t2,
             const T3 & t3,
             const T4 & t4,
             const T5 & t5,
             const T6 & t6,
             const T7 & t7,
             const T8 & t8,
             const T9 & t9,
             const T10 & t10,
             const T11 & t11,
             const T12 & t12,
             const T13 & t13,
             const T14 & t14,
             const T15 & t15,
             const T16 & t16,
             const T17 & t17,
             const T18 & t18,
             const T19 & t19,
             const T20 & t20,
             const T21 & t21,
             const T22 & t22,
             const T23 & t23,
             const T24 & t24,
             const T25 & t25,
             const T26 & t26,
             const T27 & t27,
             const T28 & t28,
             const T29 & t29,
             const T30 & t30,
             const T31 & t31,
             const T32 & t32,
             const T33 & t33,
             const T34 & t34,
             const T35 & t35,
             const T36 & t36,
             const T37 & t37,
             const T38 & t38,
             const T39 & t39,
             const T40 & t40,
             const T41 & t41,
             const T42 & t42,
             const T43 & t43,
             const T44 & t44,
             const T45 & t45,
             const T46 & t46,
             const T47 & t47,
             const T48 & t48,
             const T49 & t49 )  [inline]
```

Definition at line 807 of file cbind.h.

References MakeBindable.

5.10.2.49 cbind() [49/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 , typename T48 , typename T49 , typename T50 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
            const T1 & t1,
            const T2 & t2,
            const T3 & t3,
            const T4 & t4,
            const T5 & t5,
            const T6 & t6,
            const T7 & t7,
            const T8 & t8,
            const T9 & t9,
            const T10 & t10,
            const T11 & t11,
            const T12 & t12,
            const T13 & t13,
            const T14 & t14,
            const T15 & t15,
            const T16 & t16,
            const T17 & t17,
            const T18 & t18,
            const T19 & t19,
            const T20 & t20,
            const T21 & t21,
            const T22 & t22,
            const T23 & t23,
            const T24 & t24,
            const T25 & t25,
            const T26 & t26,
            const T27 & t27,
            const T28 & t28,
            const T29 & t29,
            const T30 & t30,
            const T31 & t31,
            const T32 & t32,
            const T33 & t33,
            const T34 & t34,
            const T35 & t35,
            const T36 & t36,
            const T37 & t37,
            const T38 & t38,
            const T39 & t39,
            const T40 & t40,
            const T41 & t41,
            const T42 & t42,
            const T43 & t43,
            const T44 & t44,
            const T45 & t45,
            const T46 & t46,
            const T47 & t47,
            const T48 & t48,
            const T49 & t49,
            const T50 & t50 )  [inline]
```

Definition at line 814 of file cbind.h.

References MakeBindable.

5.10.2.50 collapse__impl()

```
template<typename Iterator >
String Rcpp::sugar::collapse__impl (
                Iterator it,
                R_xlen_t n )  [inline]
```

Definition at line 28 of file collapse.h.

Referenced by Rcpp::collapse().

5.10.2.51 EmpiricalSample() [1/2]

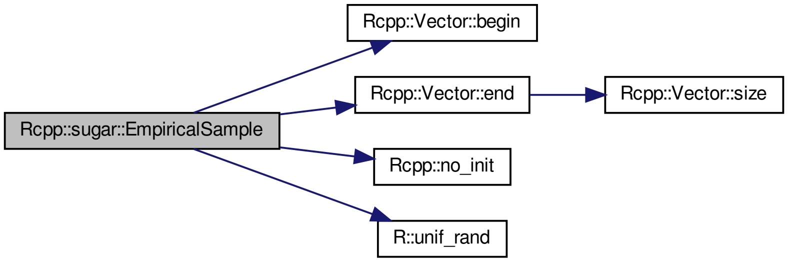
```
Vector<INTSXP> Rcpp::sugar::EmpiricalSample (
                int n,
                int size,
                bool replace,
                bool one_based )  [inline]
```

Definition at line 337 of file sample.h.

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

Referenced by Rcpp::sample().

Here is the call graph for this function:



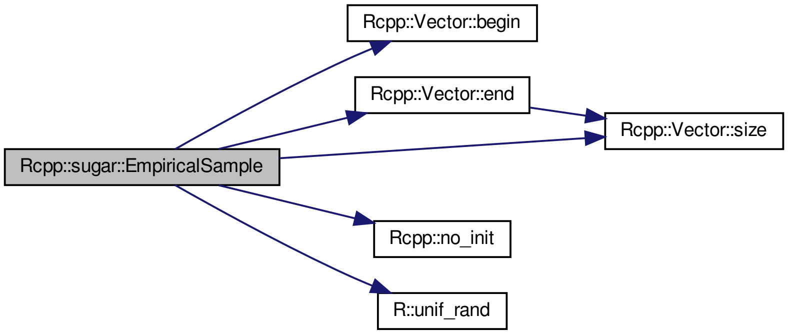
5.10.2.52 EmpiricalSample() [2/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::EmpiricalSample (
             int size,
             bool replace,
             const Vector< RTYPE > & ref )  [inline]
```

Definition at line 367 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp↩ ::no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif_rand().

Here is the call graph for this function:



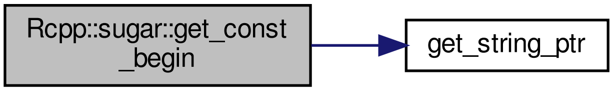
5.10.2.53 get_const_begin() [1/2]

```
template<>
SEXP* Rcpp::sugar::get_const_begin (
            const CharacterVector & obj )  [inline]
```

Definition at line 143 of file iterator.h.

References get_string_ptr().

Here is the call graph for this function:



5.10.2.54 get_const_begin() [2/2]

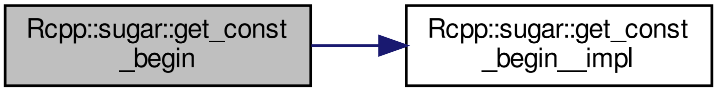
```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_begin (
            const T & obj )  [inline]
```

Definition at line 138 of file iterator.h.

References get_const_begin__impl().

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

Here is the call graph for this function:



5.10.2.55 get_const_begin__impl() [1/2]

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_begin__impl (
            const T & obj,
            Rcpp::traits::false_type ) [inline]
```

Definition at line 130 of file iterator.h.

5.10.2.56 get_const_begin__impl() [2/2]

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_begin__impl (
            const T & obj,
            Rcpp::traits::true_type ) [inline]
```

Definition at line 126 of file iterator.h.

Referenced by get_const_begin().

5.10.2.57 get_const_end()

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_end (
            const T & obj ) [inline]
```

Definition at line 148 of file iterator.h.

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

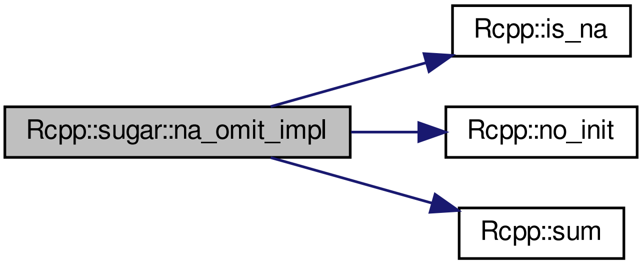
5.10.2.58 na_omit_impl() [1/2]

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

Definition at line 29 of file na_omit.h.

References Rcpp::is_na(), Rcpp::no_init(), and Rcpp::sum().

Here is the call graph for this function:



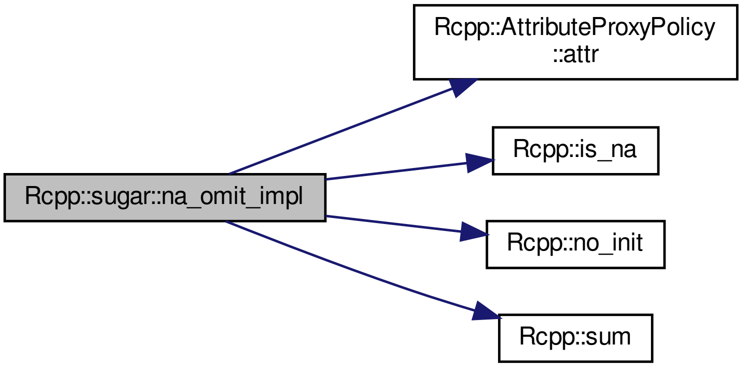
5.10.2.59 na_omit_impl() [2/2]

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

Definition at line 44 of file na_omit.h.

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

Here is the call graph for this function:



5.10.2.60 Normalize()

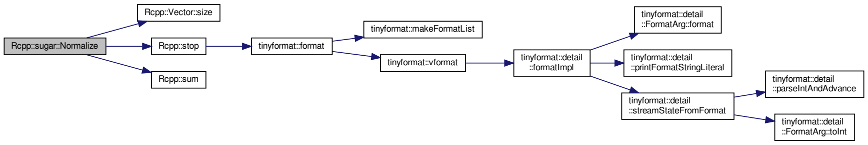
```
void Rcpp::sugar::Normalize (
            Vector< REALSXP > & p,
            int require_k,
            bool replace )  [inline]
```

Definition at line 57 of file sample.h.

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

Referenced by Rcpp::sample().

Here is the call graph for this function:


Rcpp::sugar::Normalize
Rcpp::Vector::size
Rcpp::stop
Rcpp::sum
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.10.2.61 SampleNoReplace() [1/2]

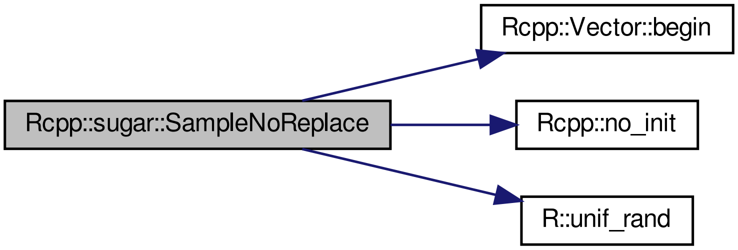
```
Vector<INTSXP> Rcpp::sugar::SampleNoReplace (
             Vector< REALSXP > & p,
             int n,
             int nans,
             bool one_based )  [inline]
```

Definition at line 255 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no_init(), and R::unif_rand().

Referenced by Rcpp::sample().

Here is the call graph for this function:



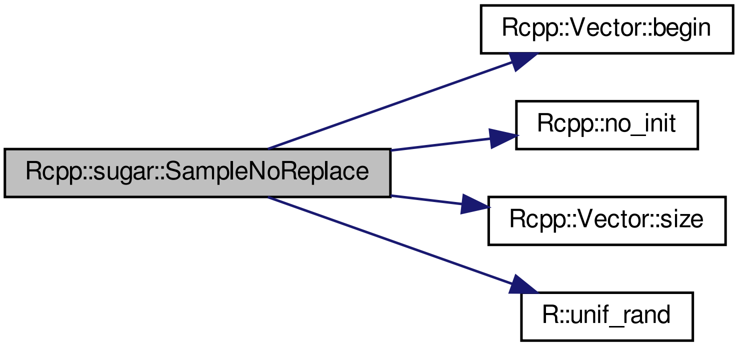
5.10.2.62 SampleNoReplace() [2/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::SampleNoReplace (
             Vector< REALSXP > & p,
             int nans,
             const Vector< RTYPE > & ref )  [inline]
```

Definition at line 295 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif_rand().

Here is the call graph for this function:



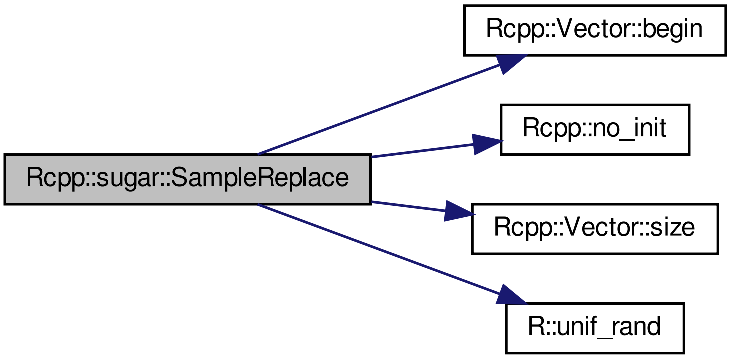
5.10.2.63 SampleReplace() [1/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::SampleReplace (
             Vector< REALSXP > & p,
             int k,
             const Vector< RTYPE > & ref )  [inline]
```

Definition at line 114 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif_rand().

Here is the call graph for this function:



5.10.2.64 SampleReplace() [2/2]

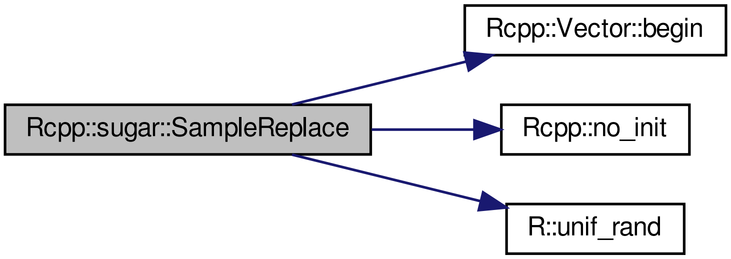
```
Vector<INTSXP> Rcpp::sugar::SampleReplace (
            Vector< REALSXP > & p,
            int n,
            int k,
            bool one_based )  [inline]
```

Definition at line 81 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no_init(), and R::unif_rand().

Referenced by Rcpp::sample().

Here is the call graph for this function:



5.10.2.65 WalkerSample() [1/2]

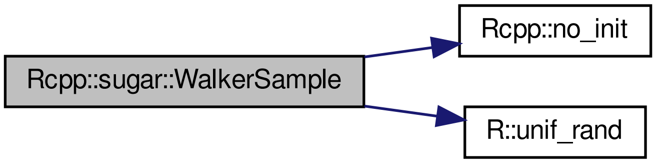
```
Vector<INTSXP> Rcpp::sugar::WalkerSample (
             const Vector< REALSXP > & p,
             int n,
             int nans,
             bool one_based )  [inline]
```

Definition at line 149 of file sample.h.

References Rcpp::no_init(), and R::unif_rand().

Referenced by Rcpp::sample().

Here is the call graph for this function:



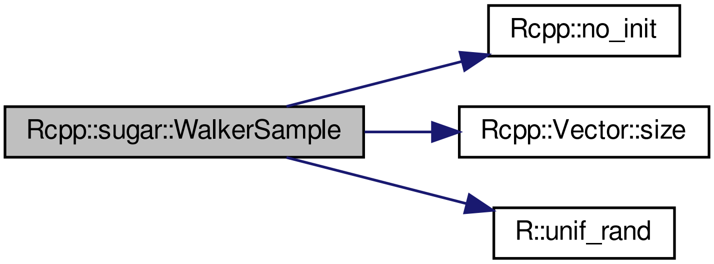
5.10.2.66 WalkerSample() [2/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::WalkerSample (
             const Vector< REALSXP > & p,
             int nans,
             const Vector< RTYPE > & ref )  [inline]
```

Definition at line 201 of file sample.h.

References Rcpp::no_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif_rand().

Here is the call graph for this function:



5.11 Rcpp::sugar::cbind_impl Namespace Reference

Namespaces

- detail

Classes

- struct cbind_sexptype_traits
- struct cbind_sexptype_traits< SEXP >
- struct cbind_storage_type
- struct cbind_storage_type< LGLSXP >
- class BindableExpression
- class ContainerBindable
- struct scalar
- class ScalarBindable
- class JoinOp
- class JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >
- class JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >
- class JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::ty
- struct matrix_return
- struct matrix_return< T, false >

Functions

- template<int RTYPE>
  ContainerBindable< RTYPE, Rcpp::Matrix< RTYPE > > MakeContainerBindable (const Rcpp::Matrix< RTYPE > &x)
- template<int RTYPE>
  ContainerBindable< RTYPE, Rcpp::Vector< RTYPE > > MakeContainerBindable (const Rcpp::Vector< RTYPE > &x)
- template<> ContainerBindable< LGLSXP, Rcpp::Matrix< LGLSXP > > MakeContainerBindable (const Rcpp::Matrix< LGLSXP > &x)
- template<> ContainerBindable< LGLSXP, Rcpp::Vector< LGLSXP > > MakeContainerBindable (const Rcpp::Vector< LGLSXP > &x)
- template<typename T >
  ScalarBindable< T > MakeScalarBindable (const T &t)
- template<int RTYPE, typename E1 , typename E2 >
  JoinOp< RTYPE, E1, E2 > operator, (const BindableExpression< RTYPE, E1 > &e1, const BindableExpression< RTYPE, E2 > &e2)

5.11.1 Function Documentation

5.11.1.1 MakeContainerBindable() [1/4]

```
template<>
ContainerBindable<LGLSXP, Rcpp::Matrix<LGLSXP> > Rcpp::sugar::cbind_impl::MakeContainerBindable
(
             const Rcpp::Matrix< LGLSXP > & x )  [inline]
```

Definition at line 343 of file cbind.h.

5.11.1.2 MakeContainerBindable() [2/4]

```
template<int RTYPE>
ContainerBindable<RTYPE, Rcpp::Matrix<RTYPE> > Rcpp::sugar::cbind_impl::MakeContainerBindable (
             const Rcpp::Matrix< RTYPE > & x )  [inline]
```

Definition at line 331 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >::operator()(), Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >::operator()(), and Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >::operator()().

5.11.1.3 MakeContainerBindable() [3/4]

```
template<>
ContainerBindable<LGLSXP, Rcpp::Vector<LGLSXP> > Rcpp::sugar::cbind_impl::MakeContainerBindable
(
             const Rcpp::Vector< LGLSXP > & x )  [inline]
```

Definition at line 349 of file cbind.h.

5.11.1.4 MakeContainerBindable() [4/4]

```
template<int RTYPE>
ContainerBindable<RTYPE, Rcpp::Vector<RTYPE> > Rcpp::sugar::cbind_impl::MakeContainerBindable (
             const Rcpp::Vector< RTYPE > & x )  [inline]
```

Definition at line 337 of file cbind.h.

5.11.1.5 MakeScalarBindable()

```
template<typename T >
ScalarBindable<T> Rcpp::sugar::cbind_impl::MakeScalarBindable (
             const T & t )  [inline]
```

Definition at line 355 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >::operator()().

5.11.1.6 operator,()

```
template<int RTYPE, typename E1 , typename E2 >
JoinOp<RTYPE, E1, E2> Rcpp::sugar::cbind_impl::operator, (
             const BindableExpression< RTYPE, E1 > & e1,
             const BindableExpression< RTYPE, E2 > & e2 )  [inline]
```

Definition at line 361 of file cbind.h.

5.12 Rcpp::sugar::cbind_impl::detail Namespace Reference

Classes

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

Functions

- template<typename T >
  Rcpp::traits::enable_if< has_stored_type< T >::value, MakeBindableCall< T, true > >::type MakeBindable (const T &t)
- template<typename T >
  Rcpp::traits::enable_if< !has_stored_type< T >::value, MakeBindableCall< T, false > >::type MakeBindable (const T &t)

5.12.1 Function Documentation

5.12.1.1 MakeBindable() [1/2]

```
template<typename T >
Rcpp::traits::enable_if< has_stored_type<T>::value, MakeBindableCall<T, true> >::type Rcpp↩
::sugar::cbind_impl::detail::MakeBindable (
             const T & t )  [inline]
```

Definition at line 446 of file cbind.h.

5.12.1.2 MakeBindable() [2/2]

```
template<typename T >
Rcpp::traits::enable_if< !has_stored_type<T>::value, MakeBindableCall<T, false> >::type Rcpp↩
::sugar::cbind_impl::detail::MakeBindable (
             const T & t )  [inline]
```

Definition at line 455 of file cbind.h.

5.13 Rcpp::sugar::detail Namespace Reference

Classes

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

Functions

- bool check_na (double x)
- bool check_na (int x)
- bool check_na (Rboolean x)
- bool check_na (SEXP x)
- bool check_na (Rcomplex x)
- void incr (double ∗lhs, double rhs)
- void incr (int ∗lhs, int rhs)
- void incr (Rcomplex ∗lhs, const Rcomplex &rhs)
- void div (double ∗lhs, R_xlen_t rhs)
- void div (Rcomplex ∗lhs, R_xlen_t rhs)
- void set_nan (double ∗x)
- void set_nan (Rcomplex ∗x)
- bool isws (const char c)
- const char ∗ trim_left (const char ∗str)
- const char ∗ trim_right (const char ∗str, R_len_t sz, std::string ∗buff)
- const char ∗ trim_both (const char ∗str, R_len_t sz, std::string ∗buff)

5.13.1 Function Documentation

5.13.1.1 check_na() [1/5]

```
bool Rcpp::sugar::detail::check_na (
             double x )  [inline]
```

Definition at line 30 of file rowSums.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

5.13.1.2 check_na() [2/5]

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

Definition at line 34 of file rowSums.h.

5.13.1.3 check_na() [3/5]

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

Definition at line 38 of file rowSums.h.

5.13.1.4 check_na() [4/5]

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

Definition at line 46 of file rowSums.h.

5.13.1.5 check_na() [5/5]

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

Definition at line 42 of file rowSums.h.

5.13.1.6 div() [1/2]

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

Definition at line 65 of file rowSums.h.

Referenced by Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

5.13.1.7 div() [2/2]

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

Definition at line 69 of file rowSums.h.

5.13.1.8 incr() [1/3]

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

Definition at line 51 of file rowSums.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

5.13.1.9 incr() [2/3]

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

Definition at line 55 of file rowSums.h.

5.13.1.10 incr() [3/3]

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

Definition at line 59 of file rowSums.h.

5.13.1.11 isws()

```
bool Rcpp::sugar::detail::isws (
            const char c )  [inline]
```

Definition at line 36 of file trimws.h.

Referenced by trim_both(), trim_left(), and trim_right().

5.13.1.12 set_nan() [1/2]

```
void Rcpp::sugar::detail::set_nan (
            double * x )  [inline]
```

Definition at line 75 of file rowSums.h.

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

5.13.1.13 set_nan() [2/2]

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

Definition at line 79 of file rowSums.h.

5.13.1.14 trim_both()

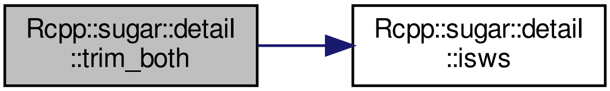
```
const char* Rcpp::sugar::detail::trim_both (
            const char * str,
            R_len_t sz,
            std::string * buff )  [inline]
```

Definition at line 66 of file trimws.h.

References isws().

Referenced by Rcpp::trimws().

Here is the call graph for this function:



5.13.1.15 trim_left()

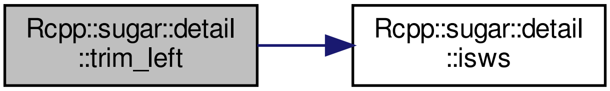
```
const char* Rcpp::sugar::detail::trim_left (
             const char * str )  [inline]
```

Definition at line 40 of file trimws.h.

References isws().

Referenced by Rcpp::trimws().

Here is the call graph for this function:



5.13.1.16 trim_right()

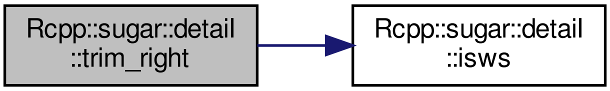
```
const char* Rcpp::sugar::detail::trim_right (
             const char * str,
             R_len_t sz,
             std::string * buff )  [inline]
```

Definition at line 52 of file trimws.h.

References isws().

Referenced by Rcpp::trimws().

Here is the call graph for this function:



5.14 Rcpp::sugar::median_detail Namespace Reference

Classes

- struct result
- struct result< INTSXP >
- struct result< STRSXP >

Functions

- template<typename T >
  bool less (T lhs, T rhs)
- template<> bool less< Rcomplex > (Rcomplex lhs, Rcomplex rhs)
- double half (double lhs)
- double half (int lhs)
- Rcomplex half (Rcomplex lhs)

5.14.1 Function Documentation

5.14.1.1 half() [1/3]

```
double Rcpp::sugar::median_detail::half (
            double lhs )  [inline]
```

Definition at line 69 of file median.h.

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

5.14.1.2 half() [2/3]

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

Definition at line 73 of file median.h.

5.14.1.3 half() [3/3]

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

Definition at line 77 of file median.h.

5.14.1.4 less()

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

Definition at line 55 of file median.h.

5.14.1.5 less< Rcomplex >()

```
template<>
bool Rcpp::sugar::median_detail::less< Rcomplex > (
             Rcomplex lhs,
             Rcomplex rhs )  [inline]
```

Definition at line 60 of file median.h.

5.15 Rcpp::traits Namespace Reference

traits used to dispatch wrap

Classes

- struct wrap_type_traits< Rcpp::Date >
- struct r_type_traits< Rcpp::Date >
- struct r_type_traits< std::pair< const std::string, Rcpp::Date > >
- struct r_sexptype_traits< Rcpp::Date >
- struct wrap_type_traits< Rcpp::Datetime >
- struct r_type_traits< Rcpp::Datetime >
- struct r_type_traits< std::pair< const std::string, Rcpp::Datetime > >
- struct r_sexptype_traits< Rcpp::Datetime >
- class ContainerExporter
- struct container_exporter< Container, int >
- struct container_exporter< Container, double >

- struct Extractor
- struct input_parameter
- struct input_parameter< T & >
- struct input_parameter< const T >
- struct input_parameter< const T & >
- struct r_type_traits< Rcpp::String >
- struct r_sexptype_traits< Rcpp::String >
- struct char_type
- struct char_type< const wchar_t ∗ >
- struct char_type< const char ∗ >
- struct enable_if
- struct enable_if< true, T >
- struct expands_to_logical__impl
- struct expands_to_logical__impl< LGLSXP >
- class _has_rtype_helper
- struct expands_to_logical
- struct __sfinae_types
- class _has_iterator_helper
- class _is_importer_helper
- class _is_generator_helper
- class _is_exporter_helper
- struct has_iterator
- struct is_importer
- struct is_exporter
- struct is_generator
- struct has_na
- struct has_na< INTSXP >
- struct has_na< REALSXP >
- struct has_na< CPLXSXP >
- struct has_na< STRSXP >
- struct has_na< LGLSXP >
- struct if_
- struct if_< false, LHS, RHS >
- struct init_type
- struct init_type< STRSXP >
- struct init_type< LGLSXP >
- struct integral_constant
- struct both
- struct is_arithmetic
- struct is_arithmetic< short >
- struct is_arithmetic< const short >
- struct is_arithmetic< unsigned short >
- struct is_arithmetic< const unsigned short >
- struct is_arithmetic< int >
- struct is_arithmetic< const int >
- struct is_arithmetic< unsigned int >
- struct is_arithmetic< const unsigned int >
- struct is_arithmetic< long >
- struct is_arithmetic< const long >
- struct is_arithmetic< unsigned long >
- struct is_arithmetic< const unsigned long >

- struct is_arithmetic< float >
- struct is_arithmetic< const float >
- struct is_arithmetic< double >
- struct is_arithmetic< const double >
- struct is_arithmetic< long double >
- struct is_arithmetic< const long double >
- struct is_bool
- struct is_bool< bool >
- struct is_bool< const bool >
- struct is_bool< volatile bool >
- struct is_const

  type properties [4.5.3].
- struct is_const< _Tp const >
- class is_convertible
- class _is_eigen_helper
- struct is_eigen_base
- struct is_module_object
- struct is_pointer
- struct is_pointer< T * >
- struct is_primitive
- struct is_reference
- struct is_reference< _Tp & >
- class _is_sugar_expression_helper
- struct is_sugar_expression
- struct is_trivial
- struct is_trivial< VECSXP >
- struct is_trivial< EXPRSXP >
- struct is_wide_string
- struct is_wide_string< const wchar_t * >
- struct is_wide_string< const char * >
- struct is_wide_string< wchar_t >
- struct is_wide_string< char >
- class _has_matrix_interface_helper
- struct matrix_interface
- struct normal_wrap_tag
- struct void_wrap_tag
- struct pointer_wrap_tag
- struct module_wrap_traits
- struct module_wrap_traits< void >
- struct module_wrap_traits< T * >
- struct needs_protection
- struct needs_protection< SEXP >
- class named_object
- class named_object< SEXP >
- struct is_named
- struct is_named< named_object< T > >
- struct is_named< Rcpp::Argument >
- struct num2type
- struct allowed_matrix_type
- struct allowed_matrix_type< true >

- class one_type
- class zero_type
- struct r_sexptype_traits
- struct r_sexptype_traits< int >
- struct r_sexptype_traits< const int >
- struct r_sexptype_traits< double >
- struct r_sexptype_traits< const double >
- struct r_sexptype_traits< bool >
- struct r_sexptype_traits< std::string >
- struct r_sexptype_traits< Rcomplex >
- struct r_sexptype_traits< Rbyte >
- struct r_sexptype_traits< unsigned int >
- struct r_sexptype_traits< float >
- struct r_sexptype_traits< long >
- struct r_sexptype_traits< unsigned long >
- struct r_sexptype_traits< long double >
- struct r_sexptype_traits< short >
- struct r_sexptype_traits< unsigned short >
- struct r_sexptype_traits< std::complex< double > >
- struct r_sexptype_traits< std::complex< float > >
- struct r_sexptype_needscast
- struct r_sexptype_needscast< int >
- struct r_sexptype_needscast< double >
- struct r_sexptype_needscast< Rcomplex >
- struct r_sexptype_needscast< Rbyte >
- struct r_type_primitive_tag
- struct r_type_string_tag
- struct r_type_generic_tag
- struct r_type_pairstring_primitive_tag
- struct r_type_pairstring_string_tag
- struct r_type_pairstring_generic_tag
- struct r_type_pair_tag
- struct r_type_module_object_pointer_tag
- struct r_type_module_object_const_pointer_tag
- struct r_type_module_object_tag
- struct r_type_module_object_reference_tag
- struct r_type_module_object_const_reference_tag
- struct r_type_enum_tag
- struct r_type_RcppString_tag
- struct r_type_traits
- struct r_type_traits< Rcpp::object< T > >
- struct r_type_traits< std::pair< const KEY, VALUE > >
- struct r_type_traits< std::pair< const std::string, T > >
- struct r_type_traits< std::pair< const std::string, int > >
- struct r_type_traits< std::pair< const std::string, const int > >
- struct r_type_traits< std::pair< const std::string, double > >
- struct r_type_traits< std::pair< const std::string, Rbyte > >
- struct r_type_traits< std::pair< const std::string, Rcomplex > >
- struct r_type_traits< std::pair< const std::string, bool > >
- struct r_type_traits< std::pair< const std::string, std::string > >
- struct r_type_traits< std::pair< const std::string, std::wstring > >

- struct r_type_traits< std::pair< const std::string, char > >
- struct r_type_traits< std::pair< const std::string, wchar_t > >
- struct r_type_traits< std::pair< const std::string, unsigned int > >
- struct r_type_traits< std::pair< const std::string, float > >
- struct r_type_traits< int >
- struct r_type_traits< const int >
- struct r_type_traits< double >
- struct r_type_traits< const double >
- struct r_type_traits< Rbyte >
- struct r_type_traits< Rcomplex >
- struct r_type_traits< bool >
- struct r_type_traits< std::string >
- struct r_type_traits< std::wstring >
- struct r_type_traits< char >
- struct r_type_traits< wchar_t >
- struct r_type_traits< unsigned int >
- struct r_type_traits< float >
- struct r_type_traits< const char ∗ >
- struct r_type_traits< const wchar_t ∗ >
- struct r_type_traits< long >
- struct r_type_traits< std::pair< const std::string, long > >
- struct r_type_traits< unsigned long >
- struct r_type_traits< std::pair< const std::string, unsigned long > >
- struct r_type_traits< long double >
- struct r_type_traits< std::pair< const std::string, long double > >
- struct r_type_traits< short >
- struct r_type_traits< std::pair< const std::string, short > >
- struct r_type_traits< unsigned short >
- struct r_type_traits< std::pair< const std::string, unsigned short > >
- struct r_type_traits< std::complex< double > >
- struct r_type_traits< std::pair< const std::string, std::complex< double > > >
- struct r_type_traits< std::complex< float > >
- struct r_type_traits< std::pair< const std::string, std::complex< float > > >
- struct remove_const

  const-volatile modifications [4.7.1].
- struct remove_const< _Tp const >
- struct remove_const_and_reference
- struct remove_reference

  reference modifications [4.7.2].
- struct remove_reference< _Tp & >
- struct result_of
- struct result_of< RESULT_TYPE(∗)(INPUT_TYPE) >
- struct result_of< RESULT_TYPE(∗)(U1, U2) >
- struct result_of< RESULT_TYPE(∗)(U1, U2, U3) >
- struct same_type
- struct same_type< T, T >
- struct storage_type
- struct storage_type< INTSXP >
- struct storage_type< REALSXP >
- struct storage_type< CPLXSXP >

- struct storage_type< RAWSXP >
- struct storage_type< LGLSXP >
- struct identity
- struct int2type
- struct un_pointer
- struct un_pointer< T ∗ >
- struct un_pointer< object< T > >
- struct wrap_type_primitive_tag
- struct wrap_type_unknown_tag
- struct wrap_type_module_object_pointer_tag
- struct wrap_type_module_object_tag
- struct wrap_type_enum_tag
- struct wrap_type_char_array
- struct wrap_type_traits
- struct wrap_type_traits< char[N]>
- struct wrap_type_traits< const char[N]>
- struct wrap_type_traits< int >
- struct wrap_type_traits< const int >
- struct wrap_type_traits< double >
- struct wrap_type_traits< Rbyte >
- struct wrap_type_traits< Rcomplex >
- struct wrap_type_traits< unsigned int >
- struct wrap_type_traits< bool >
- struct wrap_type_traits< std::string >
- struct wrap_type_traits< std::wstring >
- struct wrap_type_traits< Rcpp::String >
- struct wrap_type_traits< char >
- struct wrap_type_traits< wchar_t >
- struct wrap_type_traits< float >
- struct wrap_type_traits< std::complex< float > >
- struct wrap_type_traits< std::complex< double > >
- struct wrap_type_traits< long >
- struct wrap_type_traits< unsigned long >
- struct wrap_type_traits< long double >
- struct wrap_type_traits< short >
- struct wrap_type_traits< unsigned short >
- struct wrap_type_traits< Rcpp::object< T > >
- struct r_vector_cache_type
- class r_vector_cache
- struct r_vector_name_proxy
- struct r_vector_proxy
- struct r_vector_const_proxy
- struct r_vector_iterator
- struct r_vector_const_iterator
- struct r_vector_element_converter
- struct r_vector_element_converter< STRSXP >
- struct r_vector_element_converter< VECSXP >
- struct r_vector_element_converter< EXPRSXP >
- struct r_vector_name_proxy< STRSXP, StoragePolicy >
- struct r_vector_name_proxy< VECSXP, StoragePolicy >
- struct r_vector_name_proxy< EXPRSXP, StoragePolicy >

- struct r_vector_proxy< STRSXP, StoragePolicy >
- struct r_vector_proxy< EXPRSXP, StoragePolicy >
- struct r_vector_proxy< VECSXP, StoragePolicy >
- struct r_vector_const_proxy< STRSXP, StoragePolicy >
- struct r_vector_const_proxy< VECSXP, StoragePolicy >
- struct r_vector_const_proxy< EXPRSXP, StoragePolicy >
- struct proxy_based_iterator
- struct r_vector_iterator< VECSXP, StoragePolicy >
- struct r_vector_iterator< EXPRSXP, StoragePolicy >
- struct r_vector_iterator< STRSXP, StoragePolicy >
- struct proxy_based_const_iterator
- struct r_vector_const_iterator< VECSXP, StoragePolicy >
- struct r_vector_const_iterator< EXPRSXP, StoragePolicy >
- struct r_vector_const_iterator< STRSXP, StoragePolicy >
- class proxy_cache
- struct r_vector_cache_type< VECSXP, StoragePolicy >
- struct r_vector_cache_type< EXPRSXP, StoragePolicy >
- struct r_vector_cache_type< STRSXP, StoragePolicy >

Typedefs

- typedef integral_constant< bool, true > true_type
- typedef integral_constant< bool, false > false_type

Functions

- template<int RTYPE>
  storage_type< RTYPE >::type get_na ()
- template<> int get_na< INTSXP > ()
- template<> int get_na< LGLSXP > ()
- template<> double get_na< REALSXP > ()
- template<> Rcomplex get_na< CPLXSXP > ()
- template<> SEXP get_na< STRSXP > ()
- template<> SEXP get_na< VECSXP > ()
- template<int RTYPE>
  bool is_finite (typename storage_type< RTYPE >::type)
- template<> bool is_finite< INTSXP > (int x)
- template<> bool is_finite< REALSXP > (double x)
- template<> bool is_finite< CPLXSXP > (Rcomplex x)
- template<> bool is_finite< STRSXP > (SEXP)
- template<> bool is_finite< LGLSXP > (int x)
- template<int RTYPE>
  bool is_infinite (typename storage_type< RTYPE >::type)
- template<> bool is_infinite< REALSXP > (double x)
- template<> bool is_infinite< CPLXSXP > (Rcomplex x)
- template<int RTYPE>
  bool is_na (typename storage_type< RTYPE >::type)
- template<> bool is_na< INTSXP > (int x)
- template<> bool is_na< REALSXP > (double x)

- template<> bool is_na< CPLXSXP > (Rcomplex x)
- template<> bool is_na< STRSXP > (SEXP x)
- template<> bool is_na< LGLSXP > (int x)
- template<int RTYPE>
  bool is_nan (typename storage_type< RTYPE >::type)
- template<> bool is_nan< REALSXP > (double x)
- template<> bool is_nan< CPLXSXP > (Rcomplex x)

5.15.1 Detailed Description

traits used to dispatch wrap

5.15.2 Typedef Documentation

5.15.2.1 false_type

```
typedef integral_constant<bool, false> Rcpp::traits::false_type
```

Definition at line 35 of file integral_constant.h.

5.15.2.2 true_type

```
typedef integral_constant<bool, true> Rcpp::traits::true_type
```

Definition at line 34 of file integral_constant.h.

5.15.3 Function Documentation

5.15.3.1 get_na()

```
template<int RTYPE>
storage_type<RTYPE>::type Rcpp::traits::get_na ( )
```

5.15.3.2 get_na< CPLXSXP >()

```
template<>
Rcomplex Rcpp::traits::get_na< CPLXSXP > ( )  [inline]
```

Definition at line 42 of file get_na.h.

5.15.3.3 get_na< INTSXP >()

```
template<>
int Rcpp::traits::get_na< INTSXP > ( )  [inline]
```

Definition at line 33 of file get_na.h.

5.15.3.4 get_na< LGLSXP >()

```
template<>
int Rcpp::traits::get_na< LGLSXP > ( )  [inline]
```

Definition at line 36 of file get_na.h.

5.15.3.5 get_na< REALSXP >()

```
template<>
double Rcpp::traits::get_na< REALSXP > ( )  [inline]
```

Definition at line 39 of file get_na.h.

Referenced by Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[ ]().

5.15.3.6 get_na< STRSXP >()

```
template<>
SEXP Rcpp::traits::get_na< STRSXP > ( )  [inline]
```

Definition at line 50 of file get_na.h.

Referenced by Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type(), Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result_type(), and Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type().

5.15.3.7 get_na< VECSXP >()

```
template<>
SEXP Rcpp::traits::get_na< VECSXP > ( ) [inline]
```

Definition at line 54 of file get_na.h.

5.15.3.8 is_finite()

```
template<int RTYPE>
bool Rcpp::traits::is_finite (
             typename storage_type< RTYPE >::type  )
```

5.15.3.9 is_finite< CPLXSXP >()

```
template<>
bool Rcpp::traits::is_finite< CPLXSXP > (
             Rcomplex x ) [inline]
```

Definition at line 44 of file is_finite.h.

5.15.3.10 is_finite< INTSXP >()

```
template<>
bool Rcpp::traits::is_finite< INTSXP > (
             int x ) [inline]
```

Definition at line 34 of file is_finite.h.

5.15.3.11 is_finite< LGLSXP >()

```
template<>
bool Rcpp::traits::is_finite< LGLSXP > (
             int x ) [inline]
```

Definition at line 54 of file is_finite.h.

5.15.3.12 is_finite< REALSXP >()

```
template<>
bool Rcpp::traits::is_finite< REALSXP > (
             double x )  [inline]
```

Definition at line 39 of file is_finite.h.

5.15.3.13 is_finite< STRSXP >()

```
template<>
bool Rcpp::traits::is_finite< STRSXP > (
             SEXP  )  [inline]
```

Definition at line 49 of file is_finite.h.

5.15.3.14 is_infinite()

```
template<int RTYPE>
bool Rcpp::traits::is_infinite (
             typename storage_type< RTYPE >::type  )
```

Definition at line 30 of file is_infinite.h.

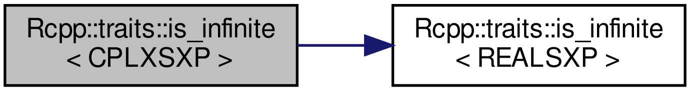
5.15.3.15 is_infinite< CPLXSXP >()

```
template<>
bool Rcpp::traits::is_infinite< CPLXSXP > (
             Rcomplex x )  [inline]
```

Definition at line 40 of file is_infinite.h.

References is_infinite< REALSXP >().

Here is the call graph for this function:



5.15.3.16 is_infinite< REALSXP >()

```
template<>
bool Rcpp::traits::is_infinite< REALSXP > (
             double x )  [inline]
```

Definition at line 35 of file is_infinite.h.

Referenced by is_infinite< CPLXSXP >().

5.15.3.17 is_na()

```
template<int RTYPE>
bool Rcpp::traits::is_na (
             typename storage_type< RTYPE >::type  )
```

Definition at line 32 of file is_na.h.

5.15.3.18 is_na< CPLXSXP >()

```
template<>
bool Rcpp::traits::is_na< CPLXSXP > (
             Rcomplex x )  [inline]
```

Definition at line 47 of file is_na.h.

Referenced by Rcpp::sugar::not_< CPLXSXP, NA >::apply(), Rcpp::sugar::unary_minus< CPLXSXP, NA >::apply(), and Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[ ]().

5.15.3.19 is_na< INTSXP >()

```
template<>
bool Rcpp::traits::is_na< INTSXP > (
             int x )  [inline]
```

Definition at line 37 of file is_na.h.

Referenced by Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[ ]().

5.15.3.20 is_na< LGLSXP >()

```
template<>
bool Rcpp::traits::is_na< LGLSXP > (
             int x )  [inline]
```

Definition at line 57 of file is_na.h.

Referenced by Rcpp::sugar::All< NA, T >::apply(), Rcpp::sugar::Any< NA, T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::negate< NA >::apply(), Rcpp::sugar::SingleLogicalResult< NA, T >::is_na(), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[](), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[]().

5.15.3.21 is_na< REALSXP >()

```
template<>
bool Rcpp::traits::is_na< REALSXP > (
             double x )  [inline]
```

Definition at line 42 of file is_na.h.

Referenced by Rcpp::sugar::not_< REALSXP, NA >::apply(), Rcpp::Date::is_na(), Rcpp::Datetime::is_na(), Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::operator()(), Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::operator()(), Rcpp::sugar::pmax_op< REALSXP, true, true >::operator()(), Rcpp::sugar::pmin_op< REALSXP, true, true >::operator()(), and Rcpp::sugar::clamp_operator< REALSXP, true >::operator()().

5.15.3.22 is_na< STRSXP >()

```
template<>
bool Rcpp::traits::is_na< STRSXP > (
             SEXP x )  [inline]
```

Definition at line 52 of file is_na.h.

Referenced by Rcpp::trimws().

5.15.3.23 is_nan()

```
template<int RTYPE>
bool Rcpp::traits::is_nan (
             typename storage_type< RTYPE >::type  )
```

Definition at line 30 of file is_nan.h.

5.15.3.24 is_nan< CPLXSXP >()

```
template<>
bool Rcpp::traits::is_nan< CPLXSXP > (
             Rcomplex x )  [inline]
```

Definition at line 40 of file is_nan.h.

5.15.3.25 is_nan< REALSXP >()

```
template<>
bool Rcpp::traits::is_nan< REALSXP > (
             double x )  [inline]
```

Definition at line 35 of file is_nan.h.

5.16 std Namespace Reference

5.17 tinyformat Namespace Reference

Namespaces

- detail

Classes

- class FormatList

Typedefs

- typedef const FormatList & FormatListRef

  Reference to type-opaque format list for passing to vformat()

Functions

- template<typename T >
  void formatValue (std::ostream &out, const char ∗, const char ∗fmtEnd, int ntrunc, const T &value)
- detail::FormatListN< 0 > makeFormatList ()
- void vformat (std::ostream &out, const char ∗fmt, FormatListRef list)
- void format (std::ostream &out, const char ∗fmt)
- std::string format (const char ∗fmt)
- void printf (const char ∗fmt)
- void printfln (const char ∗fmt)

5.17.1 Typedef Documentation

5.17.1.1 FormatListRef

```
typedef const FormatList& tinyformat::FormatListRef
```

Reference to type-opaque format list for passing to vformat()

Definition at line 867 of file tinyformat.h.

5.17.2 Function Documentation

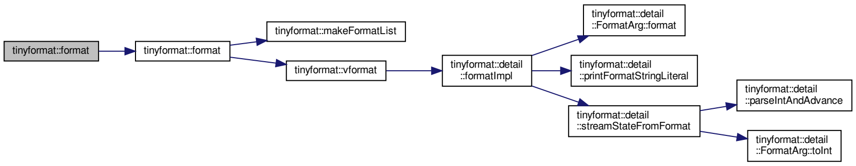
5.17.2.1 format() [1/2]

```
std::string tinyformat::format (
             const char * fmt )  [inline]
```

Definition at line 1001 of file tinyformat.h.

References format().

Here is the call graph for this function:


tinyformat::format
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.17.2.2 format() [2/2]

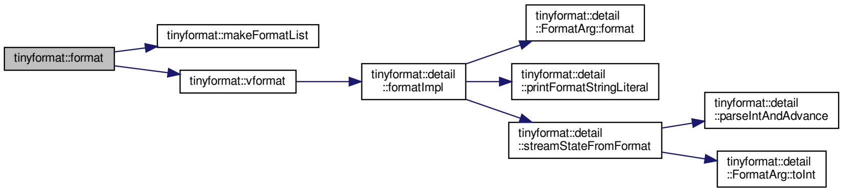
```
void tinyformat::format (
             std::ostream & out,
             const char * fmt ) [inline]
```

Definition at line 996 of file tinyformat.h.

References makeFormatList(), and vformat().

Referenced by format(), printf(), printfln(), Rcpp::sprintf(), Rcpp::stop(), and Rcpp::warning().

Here is the call graph for this function:


tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

5.17.2.3 formatValue()

```
template<typename T >
void tinyformat::formatValue (
             std::ostream & out,
             const char * ,
             const char * fmtEnd,
             int ntrunc,
             const T & value ) [inline]
```

Format a value into a stream, delegating to operator<< by default.

Users may override this for their own types. When this function is called, the stream flags will have been modified according to the format string. The format specification is provided in the range [fmtBegin, fmtEnd). For truncating conversions, ntrunc is set to the desired maximum number of characters, for example "%.7s" calls formatValue with ntrunc = 7.

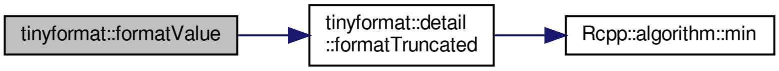
By default, formatValue() uses the usual stream insertion operator operator<< to format the type T, with special cases for the c and p conversions.

Definition at line 312 of file tinyformat.h.

References tinyformat::detail::formatTruncated().

Referenced by tinyformat::detail::FormatArg::formatImpl().

Here is the call graph for this function:



5.17.2.4 makeFormatList()

```
detail::FormatListN<0> tinyformat::makeFormatList ( )  [inline]
```

Definition at line 935 of file tinyformat.h.

Referenced by format().

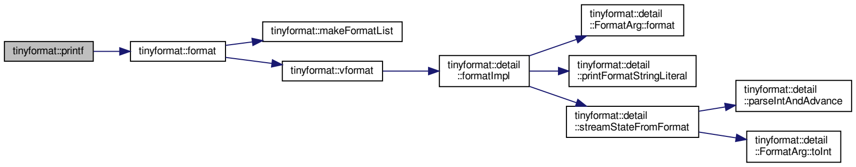
5.17.2.5 printf()

```
void tinyformat::printf (
            const char * fmt )  [inline]
```

Definition at line 1008 of file tinyformat.h.

References format().

Here is the call graph for this function:



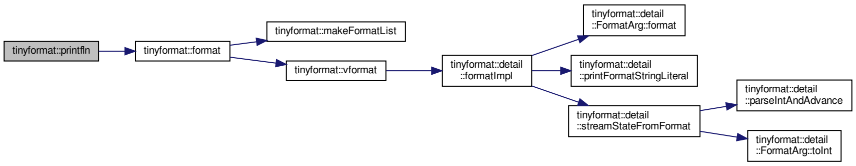
5.17.2.6 printfln()

```
void tinyformat::printfln (
             const char * fmt )  [inline]
```

Definition at line 1013 of file tinyformat.h.

References format().

Here is the call graph for this function:


tinyformat::printfln
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.17.2.7 vformat()

```
void tinyformat::vformat (
             std::ostream & out,
             const char * fmt,
             FormatListRef list )  [inline]
```

Format list of arguments to the stream according to the given format string.

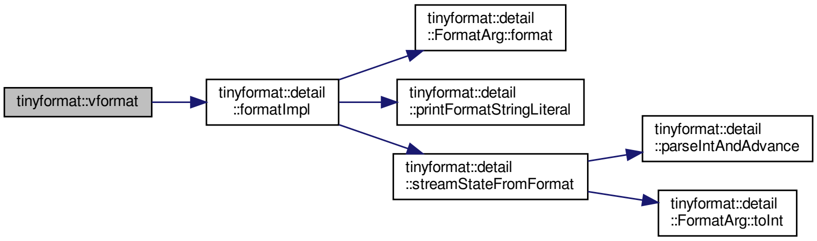
The name vformat() is chosen for the semantic similarity to vprintf(): the list of format arguments is held in a single function argument.

Definition at line 954 of file tinyformat.h.

References tinyformat::detail::formatImpl(), tinyformat::FormatList::m_formatters, and tinyformat::FormatList::m_N.

Referenced by format().

Here is the call graph for this function:


tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

5.18 tinyformat::detail Namespace Reference

Classes

- struct is_convertible
- struct is_wchar
- struct is_wchar< wchar_t * >
- struct is_wchar< const wchar_t * >
- struct is_wchar< const wchar_t[n]>
- struct is_wchar< wchar_t[n]>
- struct formatValueAsType
- struct formatValueAsType< T, fmtT, true >
- struct convertToInt
- struct convertToInt< T, true >
- class FormatArg
- class FormatListN
- class FormatListN< 0 >

Functions

- template<typename T >
  void formatTruncated (std::ostream &out, const T &value, int ntrunc)
- int parseIntAndAdvance (const char *&c)
- const char * printFormatStringLiteral (std::ostream &out, const char *fmt)
- const char * streamStateFromFormat (std::ostream &out, bool &spacePadPositive, int &ntrunc, const char *fmtStart, const detail::FormatArg *formatters, int &argIndex, int numFormatters)
- void formatImpl (std::ostream &out, const char *fmt, const detail::FormatArg *formatters, int numFormatters)

5.18.1 Function Documentation

5.18.1.1 formatImpl()

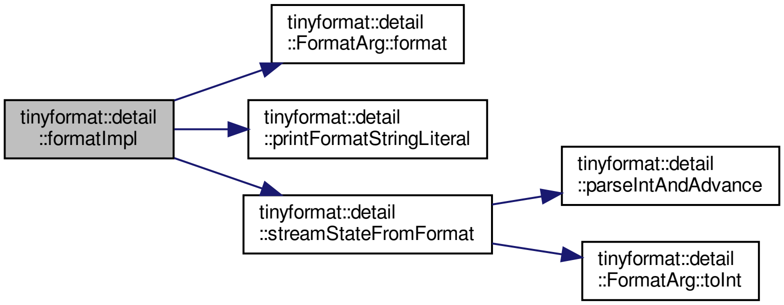
```
void tinyformat::detail::formatImpl (
            std::ostream & out,
            const char * fmt,
            const detail::FormatArg * formatters,
            int numFormatters )  [inline]
```

Definition at line 785 of file tinyformat.h.

References tinyformat::detail::FormatArg::format(), printFormatStringLiteral(), streamStateFromFormat(), and TINYFORMAT_ERROR.

Referenced by tinyformat::vformat().

Here is the call graph for this function:



5.18.1.2 formatTruncated()

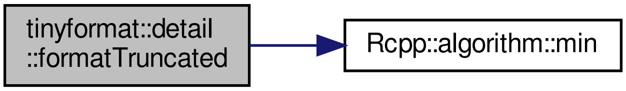
```
template<typename T >
void tinyformat::detail::formatTruncated (
             std::ostream & out,
             const T & value,
             int ntrunc )  [inline]
```

Definition at line 271 of file tinyformat.h.

References Rcpp::algorithm::min().

Referenced by tinyformat::formatValue().

Here is the call graph for this function:



5.18.1.3 parseIntAndAdvance()

```
int tinyformat::detail::parseIntAndAdvance (
             const char *& c )  [inline]
```

Definition at line 546 of file tinyformat.h.

Referenced by streamStateFromFormat().

5.18.1.4 printFormatStringLiteral()

```
const char* tinyformat::detail::printFormatStringLiteral (
             std::ostream & out,
             const char * fmt )  [inline]
```

Definition at line 560 of file tinyformat.h.

Referenced by formatImpl().

5.18.1.5 streamStateFromFormat()

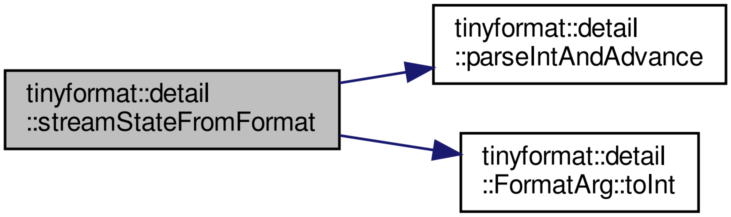
```
const char* tinyformat::detail::streamStateFromFormat (
             std::ostream & out,
             bool & spacePadPositive,
             int & ntrunc,
             const char * fmtStart,
             const detail::FormatArg * formatters,
             int & argIndex,
             int numFormatters )  [inline]
```

Definition at line 594 of file tinyformat.h.

References parseIntAndAdvance(), TINYFORMAT_ERROR, and tinyformat::detail::FormatArg::toInt().

Referenced by formatImpl().

Here is the call graph for this function:



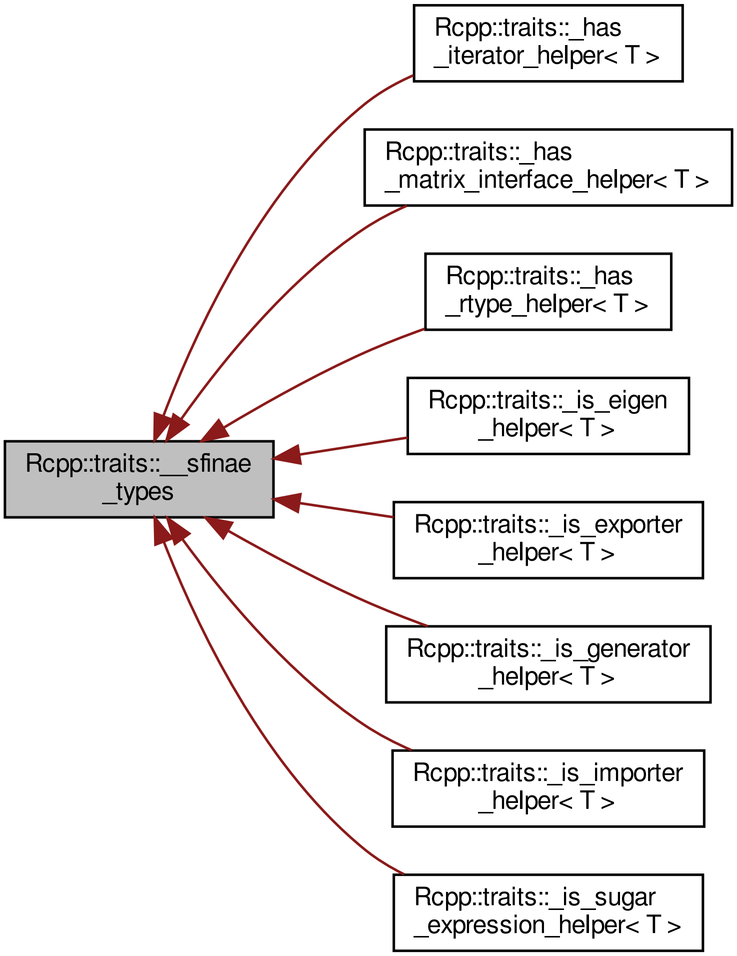
Chapter 6

Class Documentation

6.1 Rcpp::traits::__sfinae_types Struct Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::__sfinae_types:


Rcpp::traits::_has_iterator_helper< T >
Rcpp::traits::_has_matrix_interface_helper< T >
Rcpp::traits::_has_rtype_helper< T >
Rcpp::traits::_is_eigen_helper< T >
Rcpp::traits::__sfinae_types
Rcpp::traits::_is_exporter_helper< T >
Rcpp::traits::_is_generator_helper< T >
Rcpp::traits::_is_importer_helper< T >
Rcpp::traits::_is_sugar_expression_helper< T >

Classes

- struct __two

Public Types

- typedef char __one

6.1.1 Detailed Description

Definition at line 34 of file has_iterator.h.

6.1.2 Member Typedef Documentation

6.1.2.1 __one

```
typedef char Rcpp::traits::__sfinae_types::__one
```

Definition at line 35 of file has_iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.2 Rcpp::traits::__sfinae_types::__two Struct Reference

```
#include <has_iterator.h>
```

Public Attributes

- char __arr [2]

6.2.1 Detailed Description

Definition at line 36 of file has_iterator.h.

6.2.2 Member Data Documentation

6.2.2.1 __arr

```
char Rcpp::traits::__sfinae_types::__two::__arr[2]
```

Definition at line 36 of file has_iterator.h.

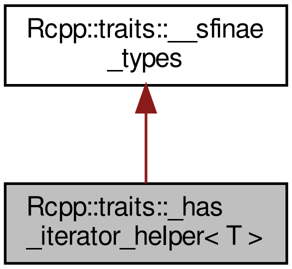
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

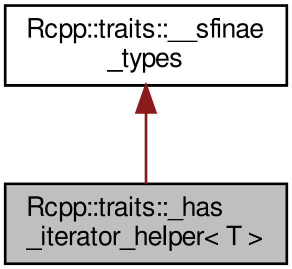
6.3 Rcpp::traits::_has_iterator_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::_has_iterator_helper< T >:



Collaboration diagram for Rcpp::traits::_has_iterator_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::iterator > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.3.1 Detailed Description

template<typename T>
class Rcpp::traits::_has_iterator_helper< T >

Definition at line 40 of file has_iterator.h.

6.3.2 Member Function Documentation

6.3.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_has_iterator_helper< T >::__test (
              ...  )  [static], [private]
```

6.3.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_has_iterator_helper< T >::__test (
             _Wrap_type< typename U::iterator > ∗  )  [static], [private]
```

6.3.3 Member Data Documentation

6.3.3.1 value

```
template<typename T >
const bool Rcpp::traits::_has_iterator_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 50 of file has_iterator.h.

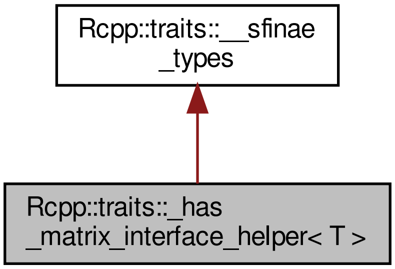
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

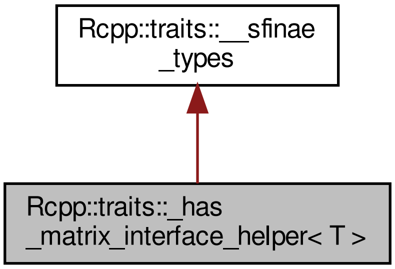
6.4 Rcpp::traits::_has_matrix_interface_helper< T > Class Template Reference

```
#include <matrix_interface.h>
```

Inheritance diagram for Rcpp::traits::_has_matrix_interface_helper< T >:



Collaboration diagram for Rcpp::traits::_has_matrix_interface_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::r_matrix_interface > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.4.1 Detailed Description

template<typename T>
class Rcpp::traits::_has_matrix_interface_helper< T >

Definition at line 33 of file matrix_interface.h.

6.4.2 Member Function Documentation

6.4.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_has_matrix_interface_helper< T >::__test (
              ...  )  [static], [private]
```

6.4.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_has_matrix_interface_helper< T >::__test (
             _Wrap_type< typename U::r_matrix_interface > ∗  )  [static], [private]
```

6.4.3 Member Data Documentation

6.4.3.1 value

```
template<typename T >
const bool Rcpp::traits::_has_matrix_interface_helper< T >::value = sizeof(__test<T>(0)) == 1
[static]
```

Definition at line 43 of file matrix_interface.h.

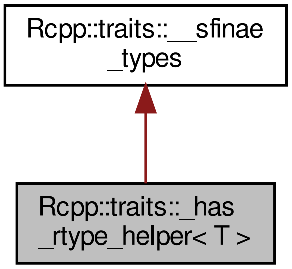
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/matrix_interface.h

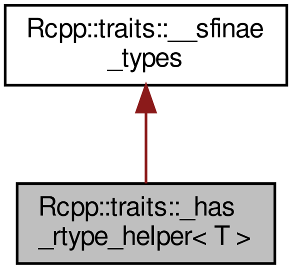
6.5 Rcpp::traits::_has_rtype_helper< T > Class Template Reference

```
#include <expands_to_logical.h>
```

Inheritance diagram for Rcpp::traits::_has_rtype_helper< T >:



Collaboration diagram for Rcpp::traits::_has_rtype_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::r_expands_to_logical > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.5.1 Detailed Description

template<typename T>
class Rcpp::traits::_has_rtype_helper< T >

Definition at line 41 of file expands_to_logical.h.

6.5.2 Member Function Documentation

6.5.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_has_rtype_helper< T >::__test (
              ... )  [static], [private]
```

6.5.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_has_rtype_helper< T >::__test (
             _Wrap_type< typename U::r_expands_to_logical > *  )  [static], [private]
```

6.5.3 Member Data Documentation

6.5.3.1 value

```
template<typename T >
const bool Rcpp::traits::_has_rtype_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 51 of file expands_to_logical.h.

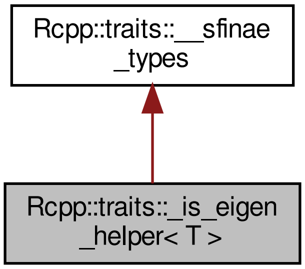
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

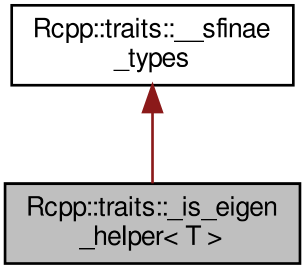
6.6 Rcpp::traits::_is_eigen_helper< T > Class Template Reference

```
#include <is_eigen_base.h>
```

Inheritance diagram for Rcpp::traits::_is_eigen_helper< T >:



Collaboration diagram for Rcpp::traits::_is_eigen_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::StorageKind > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.6.1 Detailed Description

template<typename T>
class Rcpp::traits::_is_eigen_helper< T >

Definition at line 29 of file is_eigen_base.h.

6.6.2 Member Function Documentation

6.6.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_eigen_helper< T >::__test (
              ... )  [static], [private]
```

6.6.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_eigen_helper< T >::__test (
             _Wrap_type< typename U::StorageKind > * )  [static], [private]
```

6.6.3 Member Data Documentation

6.6.3.1 value

```
template<typename T >
const bool Rcpp::traits::_is_eigen_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 39 of file is_eigen_base.h.

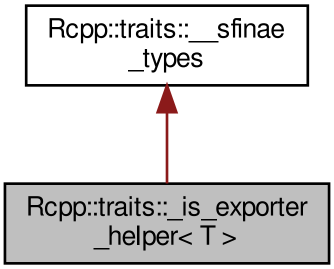
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/is_eigen_base.h

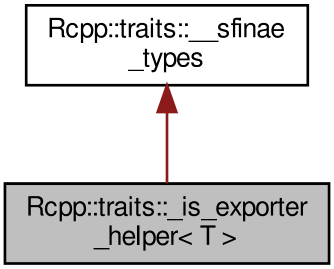
6.7 Rcpp::traits::_is_exporter_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::_is_exporter_helper< T >:


Rcpp::traits::__sfinae_types
Rcpp::traits::_is_exporter_helper< T >

Collaboration diagram for Rcpp::traits::_is_exporter_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::r_export_type > *)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.7.1 Detailed Description

template<typename T>
class Rcpp::traits::_is_exporter_helper< T >

Definition at line 83 of file has_iterator.h.

6.7.2 Member Function Documentation

6.7.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_exporter_helper< T >::__test (
              ...  )  [static], [private]
```

6.7.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_exporter_helper< T >::__test (
             _Wrap_type< typename U::r_export_type > *  )  [static], [private]
```

6.7.3 Member Data Documentation

6.7.3.1 value

```
template<typename T >
const bool Rcpp::traits::_is_exporter_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 93 of file has_iterator.h.

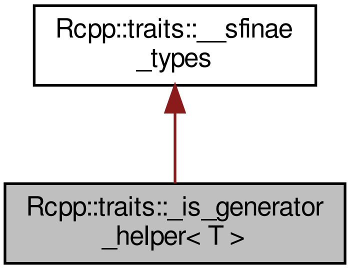
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

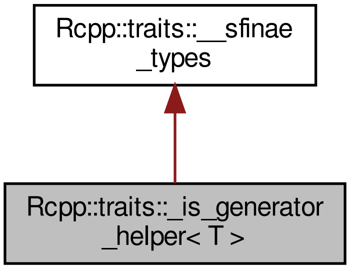
6.8 Rcpp::traits::_is_generator_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::_is_generator_helper< T >:


Rcpp::traits::__sfinae_types
Rcpp::traits::_is_generator_helper< T >

Collaboration diagram for Rcpp::traits::_is_generator_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::r_generator > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.8.1 Detailed Description

template<typename T>
class Rcpp::traits::_is_generator_helper< T >

Definition at line 68 of file has_iterator.h.

6.8.2 Member Function Documentation

6.8.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_generator_helper< T >::__test (
              ... )  [static], [private]
```

6.8.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_generator_helper< T >::__test (
             _Wrap_type< typename U::r_generator > * )  [static], [private]
```

6.8.3 Member Data Documentation

6.8.3.1 value

```
template<typename T >
const bool Rcpp::traits::_is_generator_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 78 of file has_iterator.h.

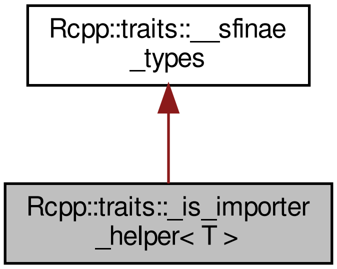
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

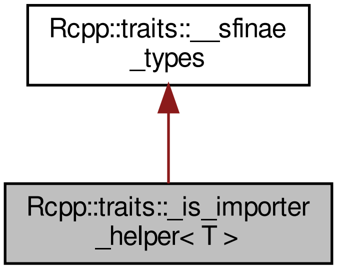
6.9 Rcpp::traits::_is_importer_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::_is_importer_helper< T >:



Collaboration diagram for Rcpp::traits::_is_importer_helper< T >:



Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::r_import_type > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.9.1 Detailed Description

template<typename T>
class Rcpp::traits::_is_importer_helper< T >

Definition at line 54 of file has_iterator.h.

6.9.2 Member Function Documentation

6.9.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_importer_helper< T >::__test (
              ...  )  [static], [private]
```

6.9.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_importer_helper< T >::__test (
             _Wrap_type< typename U::r_import_type > *  )  [static], [private]
```

6.9.3 Member Data Documentation

6.9.3.1 value

```
template<typename T >
const bool Rcpp::traits::_is_importer_helper< T >::value = sizeof(__test<T>(0)) == 1  [static]
```

Definition at line 64 of file has_iterator.h.

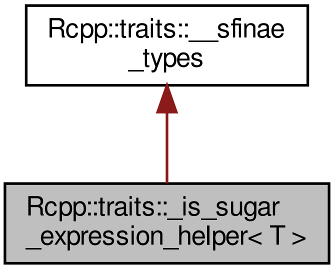
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

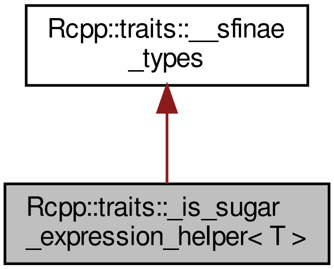
6.10 Rcpp::traits::_is_sugar_expression_helper< T > Class Template Reference

```
#include <is_sugar_expression.h>
```

Inheritance diagram for Rcpp::traits::_is_sugar_expression_helper< T >:


Rcpp::traits::__sfinae_types
Rcpp::traits::_is_sugar_expression_helper< T >

Collaboration diagram for Rcpp::traits::_is_sugar_expression_helper< T >:


Rcpp::traits::__sfinae_types
Rcpp::traits::_is_sugar_expression_helper< T >

Classes

- struct _Wrap_type

Static Public Attributes

- static const bool value = sizeof(__test<T>(0)) == 1

Static Private Member Functions

- template<typename U >
  static __one __test (_Wrap_type< typename U::rcpp_sugar_expression > ∗)
- template<typename U >
  static __two __test (...)

Additional Inherited Members

6.10.1 Detailed Description

template<typename T>
class Rcpp::traits::_is_sugar_expression_helper< T >

Definition at line 29 of file is_sugar_expression.h.

6.10.2 Member Function Documentation

6.10.2.1 __test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_sugar_expression_helper< T >::__test (
              ...  )  [static], [private]
```

6.10.2.2 __test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_sugar_expression_helper< T >::__test (
             _Wrap_type< typename U::rcpp_sugar_expression > *  )  [static], [private]
```

6.10.3 Member Data Documentation

6.10.3.1 value

```
template<typename T >
const bool Rcpp::traits::_is_sugar_expression_helper< T >::value = sizeof(__test<T>(0)) == 1
[static]
```

Definition at line 39 of file is_sugar_expression.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/is_sugar_expression.h

6.11 Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U > Struct Template Reference

6.11.1 Detailed Description

template<typename T>
template<typename U>
struct Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U >

Definition at line 41 of file has_iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.12 Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U > Struct Template Reference

6.12.1 Detailed Description

template<typename T>
template<typename U>
struct Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U >

Definition at line 34 of file matrix_interface.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/matrix_interface.h

6.13 Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U > Struct Template Reference

6.13.1 Detailed Description

template<typename T>
template<typename U>
struct Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U >

Definition at line 42 of file expands_to_logical.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

6.14 Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U > Struct Template Reference

6.14.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U >
```

Definition at line 30 of file is_eigen_base.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_eigen_base.h

6.15 Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U > Struct Template Reference

6.15.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U >
```

Definition at line 84 of file has_iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.16 Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U > Struct Template Reference

6.16.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U >
```

Definition at line 69 of file has_iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.17 Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U > Struct Template Reference

6.17.1 Detailed Description

template<typename T>
template<typename U>
struct Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U >

Definition at line 55 of file has_iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

6.18 Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U > Struct Template Reference

6.18.1 Detailed Description

template<typename T>
template<typename U>
struct Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U >

Definition at line 30 of file is_sugar_expression.h.

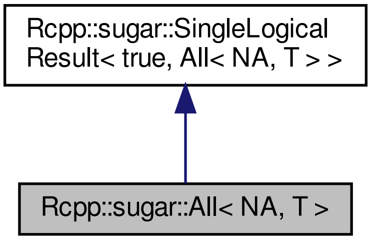
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_sugar_expression.h

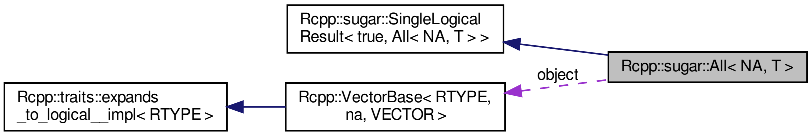
6.19 Rcpp::sugar::All< NA, T > Class Template Reference

`#include <all.h>`

Inheritance diagram for Rcpp::sugar::All< NA, T >:


Rcpp::sugar::SingleLogical Result< true, All< NA, T > >
Rcpp::sugar::All< NA, T >

Collaboration diagram for Rcpp::sugar::All< NA, T >:


Rcpp::sugar::SingleLogical
Result< true, All< NA, T > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
na, VECTOR >
object
Rcpp::sugar::All< NA, T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, NA, T > VEC_TYPE
- typedef SingleLogicalResult< true, All< NA, T > > PARENT

Public Member Functions

- All (const VEC_TYPE &t)
- void apply ()

Private Attributes

- const VEC_TYPE & object

Additional Inherited Members

6.19.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::All< NA, T >

Definition at line 29 of file all.h.

6.19.2 Member Typedef Documentation

6.19.2.1 PARENT

```
template<bool NA, typename T >
typedef SingleLogicalResult< true, All<NA,T> > Rcpp::sugar::All< NA, T >::PARENT
```

Definition at line 32 of file all.h.

6.19.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::All< NA, T >::VEC_TYPE
```

Definition at line 31 of file all.h.

6.19.3 Constructor & Destructor Documentation

6.19.3.1 All()

```
template<bool NA, typename T >
Rcpp::sugar::All< NA, T >::All (
             const VEC_TYPE & t )  [inline]
```

Definition at line 33 of file all.h.

6.19.4 Member Function Documentation

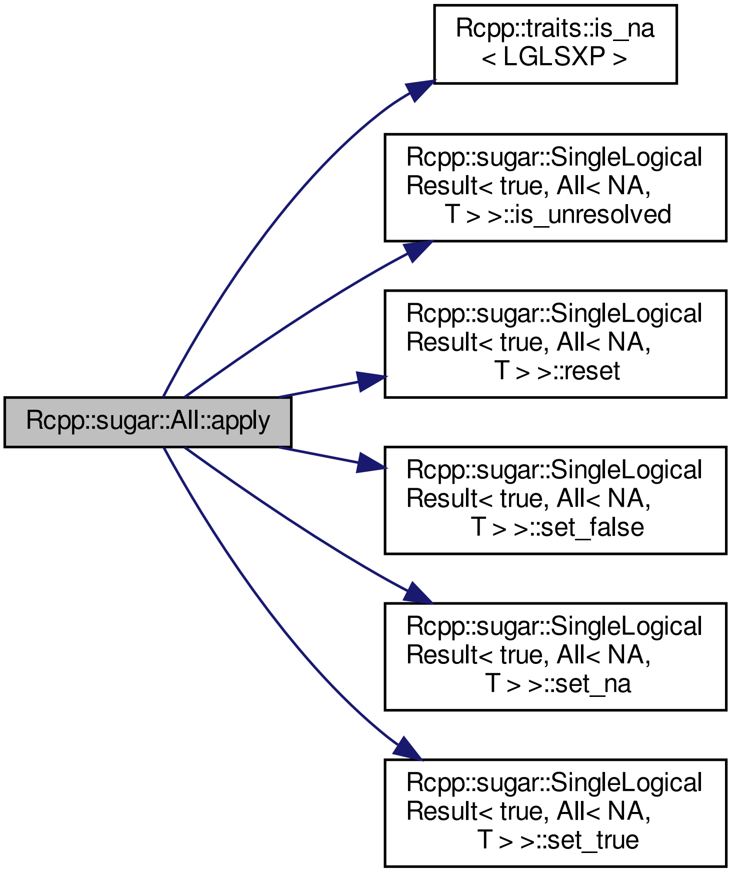
6.19.4.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::All< NA, T >::apply ( )  [inline]
```

Definition at line 35 of file all.h.

References Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::is_unresolved(), Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::reset(), Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_false(), Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_na(), and Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_true().

Here is the call graph for this function:


Rcpp::traits::is_na < LGLSXP >
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::is_unresolved
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::reset
Rcpp::sugar::All::apply
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::set_false
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::set_na
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::set_true

6.19.5 Member Data Documentation

6.19.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::All< NA, T >::object  [private]
```

Definition at line 54 of file all.h.

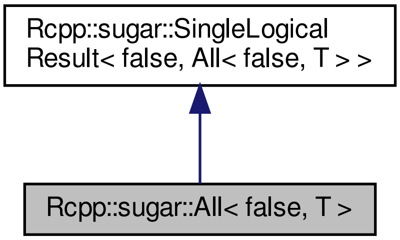
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/all.h

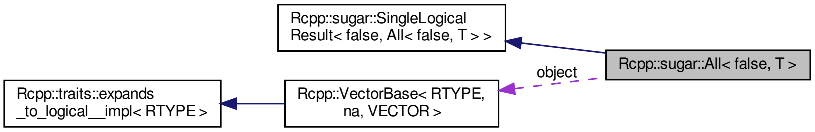
6.20 Rcpp::sugar::All< false, T > Class Template Reference

```
#include <all.h>
```

Inheritance diagram for Rcpp::sugar::All< false, T >:


Rcpp::sugar::SingleLogical
Result< false, All< false, T > >
Rcpp::sugar::All< false, T >

Collaboration diagram for Rcpp::sugar::All< false, T >:


Rcpp::sugar::SingleLogical
Result< false, All< false, T > >
Rcpp::sugar::All< false, T >
object
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, T > VEC_TYPE
- typedef SingleLogicalResult< false, All< false, T > > PARENT

Public Member Functions

- All (const VEC_TYPE &t)
- void apply ()

Private Attributes

- const VEC_TYPE & object

Additional Inherited Members

6.20.1 Detailed Description

template<typename T>
class Rcpp::sugar::All< false, T >

Definition at line 60 of file all.h.

6.20.2 Member Typedef Documentation

6.20.2.1 PARENT

```
template<typename T >
typedef SingleLogicalResult< false, All<false,T> > Rcpp::sugar::All< false, T >::PARENT
```

Definition at line 63 of file all.h.

6.20.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<LGLSXP,false,T> Rcpp::sugar::All< false, T >::VEC_TYPE
```

Definition at line 62 of file all.h.

6.20.3 Constructor & Destructor Documentation

6.20.3.1 All()

```
template<typename T >
Rcpp::sugar::All< false, T >::All (
             const VEC_TYPE & t )  [inline]
```

Definition at line 64 of file all.h.

6.20.4 Member Function Documentation

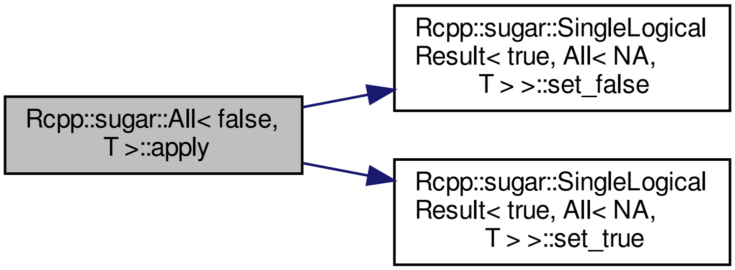
6.20.4.1 apply()

```
template<typename T >
void Rcpp::sugar::All< false, T >::apply ( )  [inline]
```

Definition at line 66 of file all.h.

References Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_false(), and Rcpp::sugar::SingleLogical↩ Result< true, All< NA, T > >::set_true().

Here is the call graph for this function:


Rcpp::sugar::All< false, T >::apply
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::set_false
Rcpp::sugar::SingleLogical Result< true, All< NA, T > >::set_true

6.20.5 Member Data Documentation

6.20.5.1 object

```
template<typename T >
const VEC_TYPE& Rcpp::sugar::All< false, T >::object  [private]
```

Definition at line 77 of file all.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/all.h

6.21 Rcpp::traits::allowed_matrix_type< bool > Struct Template Reference

6.21.1 Detailed Description

template<bool>
struct Rcpp::traits::allowed_matrix_type< bool >

Definition at line 29 of file one_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/one_type.h

6.22 Rcpp::traits::allowed_matrix_type< true > Struct Reference

```
#include <one_type.h>
```

6.22.1 Detailed Description

Definition at line 32 of file one_type.h.

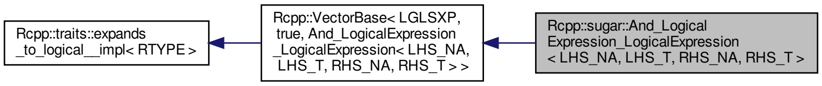
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/one_type.h

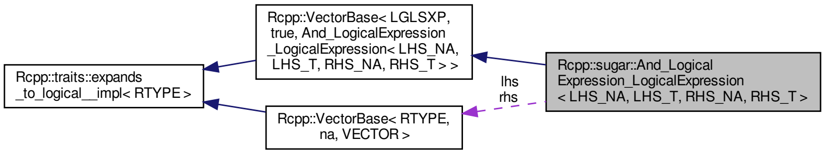
6.23 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS↩ _T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::And_Logical Expression_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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP,
true, And_LogicalExpression
_LogicalExpression< LHS_NA,
LHS_T, RHS_NA, RHS_T > >
Rcpp::VectorBase< RTYPE,
na, VECTOR >
lhs
rhs
Rcpp::sugar::And_Logical
Expression_LogicalExpression
< LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > RHS_TYPE

Public Member Functions

- And_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.23.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 205 of file and.h.

6.23.2 Member Typedef Documentation

6.23.2.1 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,LHS_NA,LHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 207 of file and.h.

6.23.2.2 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,RHS_NA,RHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 208 of file and.h.

6.23.3 Constructor & Destructor Documentation

6.23.3.1 And_LogicalExpression_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::And_LogicalExpression_Logic
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 210 of file and.h.

6.23.4 Member Function Documentation

6.23.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 212 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

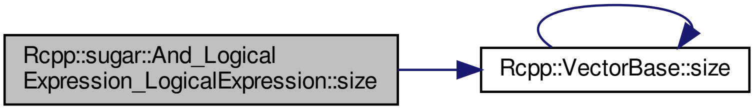
6.23.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::size ( ) const  [inline]
```

Definition at line 217 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.23.5 Member Data Documentation

6.23.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS↩
_T >::lhs  [private]
```

Definition at line 220 of file and.h.

Referenced by Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_↩T >::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_↩T >::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >↩::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp↩::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::And↩_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::And_Logical↩Expression_LogicalExpression< false, LHS_T, false, RHS_T >::size().

6.23.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS↩
_T >::rhs  [private]
```

Definition at line 221 of file and.h.

Referenced by Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_↩ T >::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_↩ T >::operator[ ](), Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >↩ ::operator[ ](), and Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >↩ ::operator[ ]().

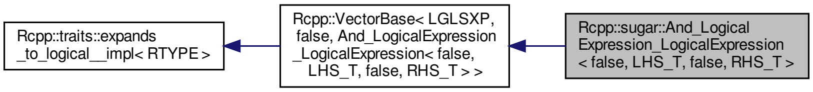
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

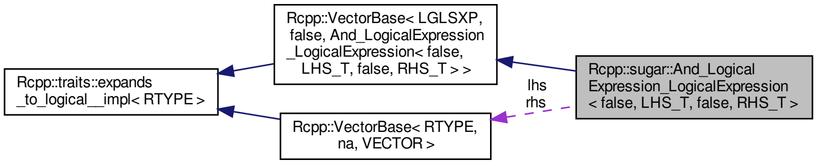
6.24 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression _LogicalExpression< false, LHS_T, false, RHS_T > >
Rcpp::sugar::And_Logical Expression_LogicalExpression < false, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression _LogicalExpression< false, LHS_T, false, RHS_T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
rhs
Rcpp::sugar::And_Logical Expression_LogicalExpression < false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, false, RHS_T > RHS_TYPE

Public Member Functions

- And_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.24.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >

Definition at line 264 of file and.h.

6.24.2 Member Typedef Documentation

6.24.2.1 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,LHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 267 of file and.h.

6.24.2.2 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,RHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 268 of file and.h.

6.24.3 Constructor & Destructor Documentation

6.24.3.1 And_LogicalExpression_LogicalExpression()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::And_LogicalExpression_Logica
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 270 of file and.h.

6.24.4 Member Function Documentation

6.24.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 272 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

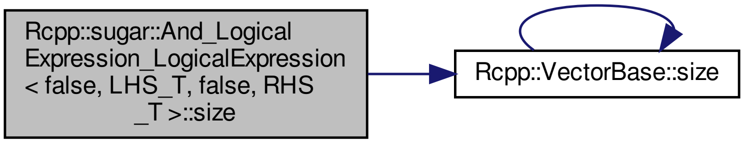
6.24.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size
( ) const  [inline]
```

Definition at line 276 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.24.5 Member Data Documentation

6.24.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 279 of file and.h.

6.24.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 280 of file and.h.

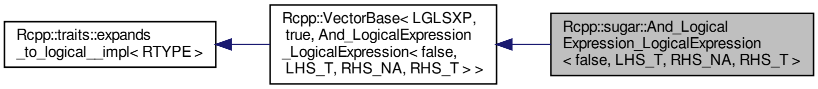
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

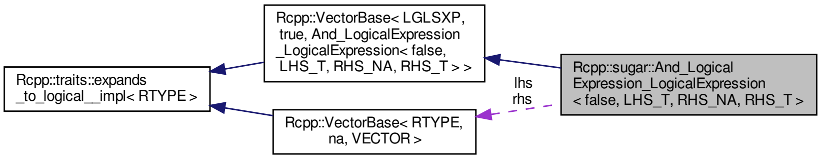
6.25 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP,
true, And_LogicalExpression
_LogicalExpression< false,
LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::And_Logical
Expression_LogicalExpression
< false, LHS_T, RHS_NA, RHS_T >

Collaboration diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP,
true, And_LogicalExpression
_LogicalExpression< false,
LHS_T, RHS_NA, RHS_T > >
Rcpp::VectorBase< RTYPE,
na, VECTOR >
lhs
rhs
Rcpp::sugar::And_Logical
Expression_LogicalExpression
< false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > RHS_TYPE

Public Member Functions

- And_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.25.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >

Definition at line 224 of file and.h.

6.25.2 Member Typedef Documentation

6.25.2.1 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,LHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 227 of file and.h.

6.25.2.2 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,RHS_NA,RHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 228 of file and.h.

6.25.3 Constructor & Destructor Documentation

6.25.3.1 And_LogicalExpression_LogicalExpression()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::And_LogicalExpression_Logica
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 230 of file and.h.

6.25.4 Member Function Documentation

6.25.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 232 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

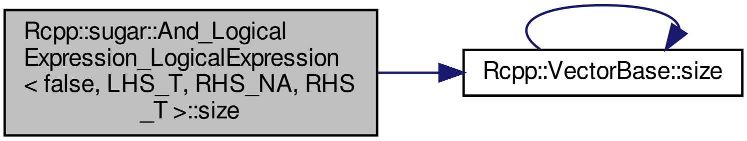
6.25.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >↩
::size ( ) const  [inline]
```

Definition at line 237 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.25.5 Member Data Documentation

6.25.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::lhs [private]
```

Definition at line 240 of file and.h.

6.25.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::rhs [private]
```

Definition at line 241 of file and.h.

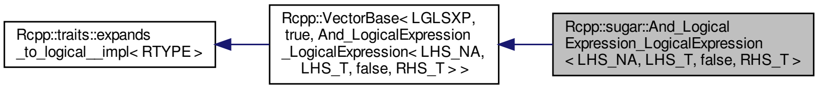
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

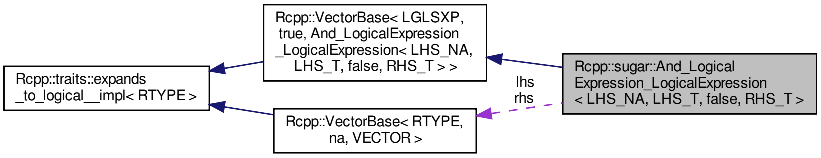
6.26 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::And_Logical Expression_LogicalExpression < LHS_NA, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
lhs
rhs
Rcpp::sugar::And_Logical Expression_LogicalExpression < LHS_NA, LHS_T, false, RHS_T >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, false, RHS_T > RHS_TYPE

Public Member Functions

- And_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.26.1 Detailed Description

template< bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >

Definition at line 244 of file and.h.

6.26.2 Member Typedef Documentation

6.26.2.1 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,LHS_NA,LHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 247 of file and.h.

6.26.2.2 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,RHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 248 of file and.h.

6.26.3 Constructor & Destructor Documentation

6.26.3.1 And_LogicalExpression_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::And_LogicalExpression_Logica
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 250 of file and.h.

6.26.4 Member Function Documentation

6.26.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 252 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

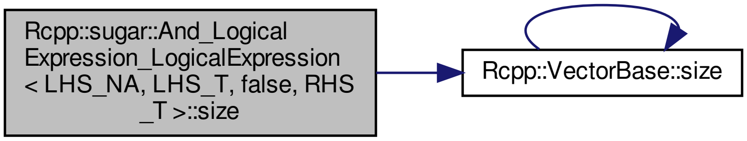
6.26.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >↩
::size ( ) const  [inline]
```

Definition at line 257 of file and.h.

References Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.26.5 Member Data Documentation

6.26.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 260 of file and.h.

6.26.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 261 of file and.h.

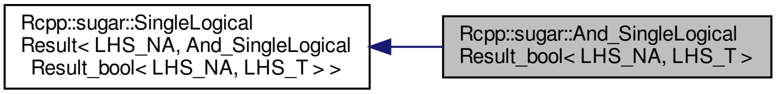
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

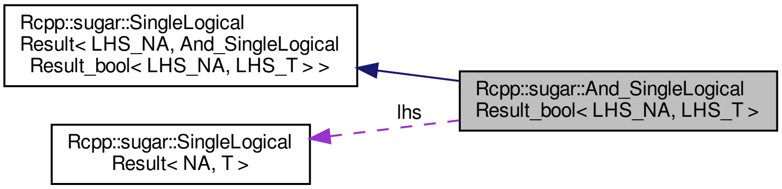
6.27 Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, And_SingleLogical Result_bool< LHS_NA, LHS_T > >
Rcpp::sugar::And_SingleLogical Result_bool< LHS_NA, LHS_T >

Collaboration diagram for Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >:


Rcpp::sugar::SingleLogical
Result< LHS_NA, And_SingleLogical
Result_bool< LHS_NA, LHS_T > >
Rcpp::sugar::SingleLogical
Result< NA, T >
lhs
Rcpp::sugar::And_SingleLogical
Result_bool< LHS_NA, LHS_T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< LHS_NA, And_SingleLogicalResult_bool< LHS_NA, LHS_T > > BASE

Public Member Functions

- And_SingleLogicalResult_bool (const LHS_TYPE &lhs_, bool rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- bool rhs

Additional Inherited Members

6.27.1 Detailed Description

template<bool LHS_NA, typename LHS_T>
class Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >

Definition at line 171 of file and.h.

6.27.2 Member Typedef Documentation

6.27.2.1 BASE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult< LHS_NA , And_SingleLogicalResult_bool<LHS_NA,LHS_T> > Rcpp::sugar::And_SingleLogica
LHS_NA, LHS_T >::BASE
```

Definition at line 182 of file and.h.

6.27.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA,
LHS_T >::LHS_TYPE
```

Definition at line 178 of file and.h.

6.27.3 Constructor & Destructor Documentation

6.27.3.1 And_SingleLogicalResult_bool()

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::And_SingleLogicalResult_bool (
            const LHS_TYPE & lhs_,
            bool rhs_ )  [inline]
```

Definition at line 184 of file and.h.

6.27.4 Member Function Documentation

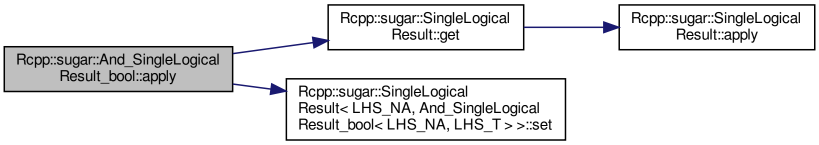
6.27.4.1 apply()

```
template<bool LHS_NA, typename LHS_T >
void Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply ( )  [inline]
```

Definition at line 187 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult< LHS_NA, And_SingleLogicalResult_bool< LHS_NA, LHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::And_SingleLogical Result_bool::apply
Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::SingleLogical Result< LHS_NA, And_SingleLogical Result_bool< LHS_NA, LHS_T > >::set

6.27.5 Member Data Documentation

6.27.5.1 lhs

```
template<bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 196 of file and.h.

Referenced by Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

6.27.5.2 rhs

```
template<bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs  [private]
```

Definition at line 197 of file and.h.

Referenced by Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

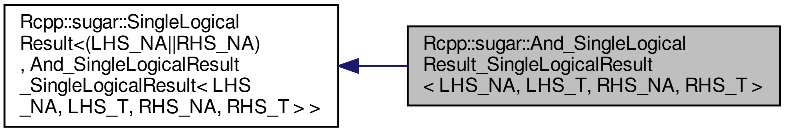
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

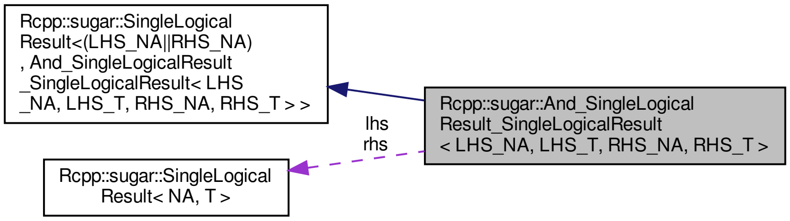
6.28 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::sugar::SingleLogical Result<(LHS_NA||RHS_NA) , And_SingleLogicalResult _SingleLogicalResult< LHS _NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::And_SingleLogical Result_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 >:


Rcpp::sugar::SingleLogical Result<(LHS_NA||RHS_NA) , And_SingleLogicalResult _SingleLogicalResult< LHS _NA, LHS_T, RHS_NA, RHS_T > >

Rcpp::sugar::SingleLogical Result< NA, T >
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< RHS_NA, RHS_T > RHS_TYPE
- typedef SingleLogicalResult<(LHS_NA||RHS_NA), And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > > BASE

Public Member Functions

- And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.28.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file and.h.

6.28.2 Member Typedef Documentation

6.28.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< (LHS_NA || RHS_NA) , And_SingleLogicalResult_SingleLogicalResult<LHS↩
_NA,LHS_T,RHS_NA,RHS_T> > Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS↩
_T, RHS_NA, RHS_T >::BASE
```

Definition at line 41 of file and.h.

6.28.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 36 of file and.h.

6.28.2.3 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 37 of file and.h.

6.28.3 Constructor & Destructor Documentation

6.28.3.1 And_SingleLogicalResult_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::And_SingleLogicalResult
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 43 of file and.h.

6.28.4 Member Function Documentation

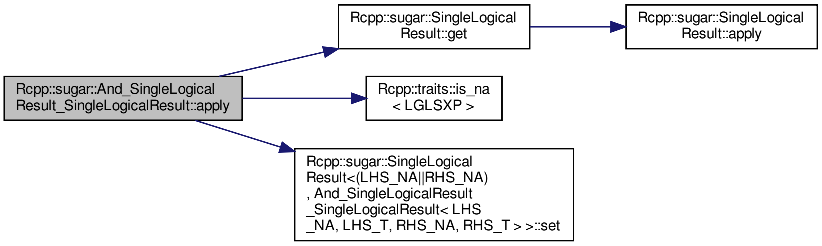
6.28.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 46 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult::apply
Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::traits::is_na < LGLSXP >
Rcpp::sugar::SingleLogical Result<(LHS_NA||RHS_NA) , And_SingleLogicalResult _SingleLogicalResult< LHS _NA, LHS_T, RHS_NA, RHS_T > >::set

6.28.5 Member Data Documentation

6.28.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::lhs  [private]
```

Definition at line 58 of file and.h.

Referenced by Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply().

6.28.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::rhs  [private]
```

Definition at line 59 of file and.h.

Referenced by Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), and Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply().

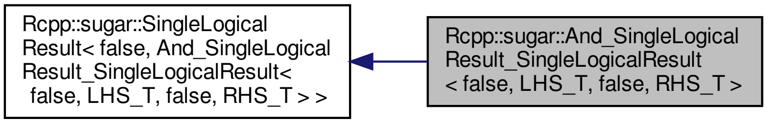
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

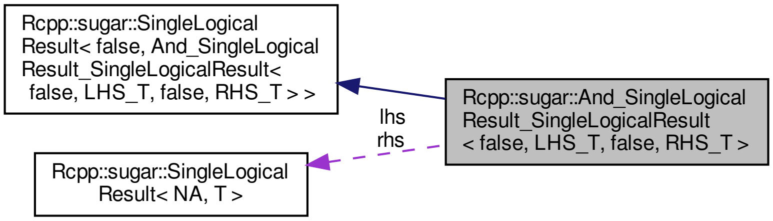
6.29 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical Result< false, And_SingleLogical Result_SingleLogicalResult< false, LHS_T, false, RHS_T > >
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < false, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical Result< false, And_SingleLogical Result_SingleLogicalResult< false, LHS_T, false, RHS_T > >
lhs
rhs
Rcpp::sugar::SingleLogical Result< NA, T >
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < false, LHS_T, false, RHS_T >

Public Types

- typedef SingleLogicalResult< false, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< false, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< false, And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > > BASE

Public Member Functions

- And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.29.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >

Definition at line 136 of file and.h.

6.29.2 Member Typedef Documentation

6.29.2.1 BASE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< false, And_SingleLogicalResult_SingleLogicalResult<false,LHS_↩
T,false,RHS_T> > Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::BASE
```

Definition at line 148 of file and.h.

6.29.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 143 of file and.h.

6.29.2.3 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 144 of file and.h.

6.29.3 Constructor & Destructor Documentation

6.29.3.1 And_SingleLogicalResult_SingleLogicalResult()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::And_SingleLogicalResult_S
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 150 of file and.h.

6.29.4 Member Function Documentation

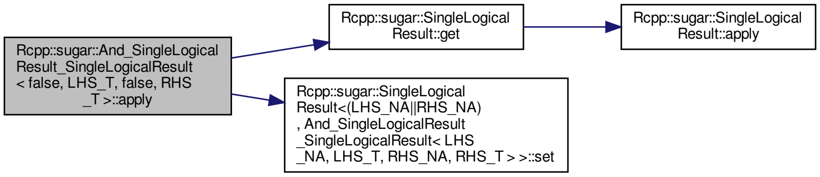
6.29.4.1 apply()

```
template<typename LHS_T , typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 153 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::And_SingleLogical
Result_SingleLogicalResult
< false, LHS_T, false, RHS
_T >::apply
Rcpp::sugar::SingleLogical
Result::get
Rcpp::sugar::SingleLogical
Result::apply
Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, And_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T > >::set

6.29.5 Member Data Documentation

6.29.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 163 of file and.h.

6.29.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 164 of file and.h.

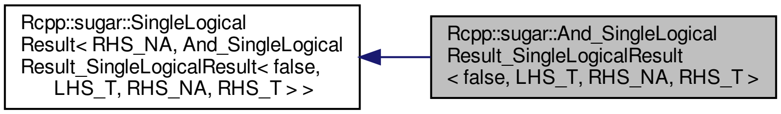
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

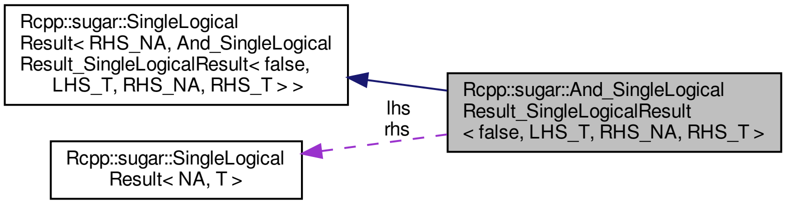
6.30 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::sugar::SingleLogical Result< RHS_NA, And_SingleLogical Result_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < false, LHS_T, RHS_NA, RHS_T >

Collaboration diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS↩_T >:


Rcpp::sugar::SingleLogical Result< RHS_NA, And_SingleLogical Result_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::SingleLogical Result< NA, T >
lhs
rhs
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef SingleLogicalResult< false, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< RHS_NA, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< RHS_NA, And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_↩ NA, RHS_T > > BASE

Public Member Functions

- And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.30.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >

Definition at line 101 of file and.h.

6.30.2 Member Typedef Documentation

6.30.2.1 BASE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< RHS_NA, And_SingleLogicalResult_SingleLogicalResult<false,LHS_↩
T,RHS_NA,RHS_T> > Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::BASE
```

Definition at line 113 of file and.h.

6.30.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file and.h.

6.30.2.3 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file and.h.

6.30.3 Constructor & Destructor Documentation

6.30.3.1 And_SingleLogicalResult_SingleLogicalResult()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::And_SingleLogicalResult_
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 115 of file and.h.

6.30.4 Member Function Documentation

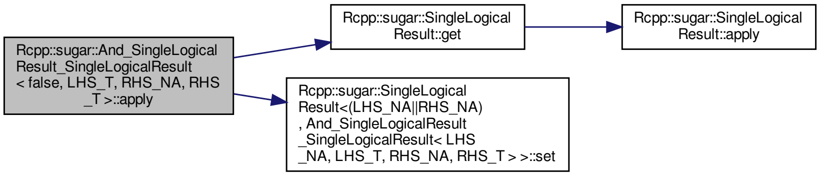
6.30.4.1 apply()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >↩
::apply ( ) [inline]
```

Definition at line 118 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::And_SingleLogical
Result_SingleLogicalResult
< false, LHS_T, RHS_NA, RHS
_T >::apply
Rcpp::sugar::SingleLogical
Result::get
Rcpp::sugar::SingleLogical
Result::apply
Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, And_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T > >::set

6.30.5 Member Data Documentation

6.30.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 129 of file and.h.

6.30.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 130 of file and.h.

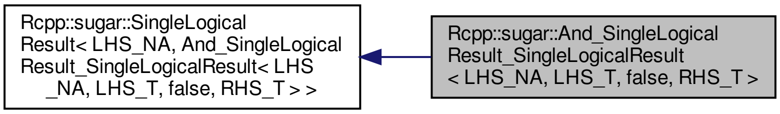
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

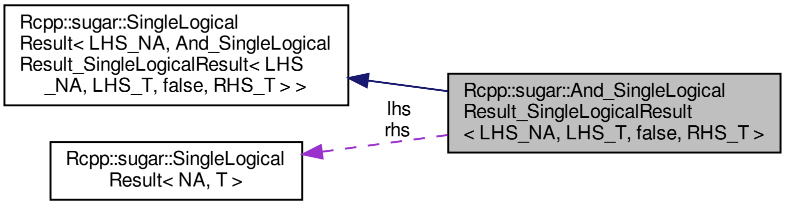
6.31 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, And_SingleLogical Result_SingleLogicalResult< LHS _NA, LHS_T, false, RHS_T > >
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < LHS_NA, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, And_SingleLogical Result_SingleLogicalResult< LHS _NA, LHS_T, false, RHS_T > >
Rcpp::sugar::SingleLogical Result< NA, T >
lhs
rhs
Rcpp::sugar::And_SingleLogical Result_SingleLogicalResult < LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< false, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< LHS_NA, And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > > BASE

Public Member Functions

- And_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.31.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >

Definition at line 65 of file and.h.

6.31.2 Member Typedef Documentation

6.31.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< LHS_NA, And_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_↩
T,false,RHS_T> > Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::BASE
```

Definition at line 77 of file and.h.

6.31.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 72 of file and.h.

6.31.2.3 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 73 of file and.h.

6.31.3 Constructor & Destructor Documentation

6.31.3.1 And_SingleLogicalResult_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::And_SingleLogicalResult_
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 79 of file and.h.

6.31.4 Member Function Documentation

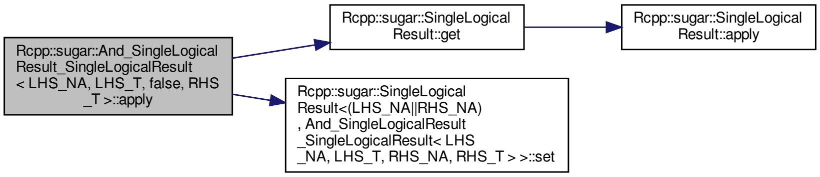
6.31.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 82 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And_SingleLogicalResult_SingleLogical↩ Result< 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:


Rcpp::sugar::And_SingleLogical
Result_SingleLogicalResult
< LHS_NA, LHS_T, false, RHS
_T >::apply
Rcpp::sugar::SingleLogical
Result::get
Rcpp::sugar::SingleLogical
Result::apply
Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, And_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T > >::set

6.31.5 Member Data Documentation

6.31.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 93 of file and.h.

6.31.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 94 of file and.h.

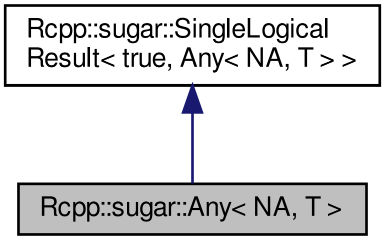
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

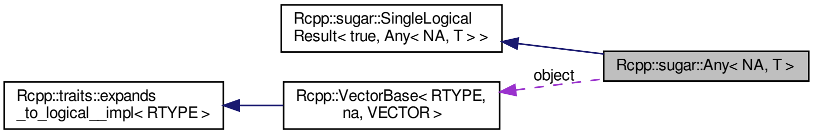
6.32 Rcpp::sugar::Any< NA, T > Class Template Reference

```
#include <any.h>
```

Inheritance diagram for Rcpp::sugar::Any< NA, T >:


Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >
Rcpp::sugar::Any< NA, T >

Collaboration diagram for Rcpp::sugar::Any< NA, T >:


Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >
Rcpp::sugar::Any< NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, NA, T > VEC_TYPE
- typedef SingleLogicalResult< true, Any< NA, T > > PARENT

Public Member Functions

- Any (const VEC_TYPE &t)
- void apply ()

Private Attributes

- const VEC_TYPE & object

Additional Inherited Members

6.32.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Any< NA, T >

Definition at line 29 of file any.h.

6.32.2 Member Typedef Documentation

6.32.2.1 PARENT

```
template<bool NA, typename T >
typedef SingleLogicalResult< true , Any<NA,T> > Rcpp::sugar::Any< NA, T >::PARENT
```

Definition at line 32 of file any.h.

6.32.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::Any< NA, T >::VEC_TYPE
```

Definition at line 31 of file any.h.

6.32.3 Constructor & Destructor Documentation

6.32.3.1 Any()

```
template<bool NA, typename T >
Rcpp::sugar::Any< NA, T >::Any (
             const VEC_TYPE & t )  [inline]
```

Definition at line 33 of file any.h.

6.32.4 Member Function Documentation

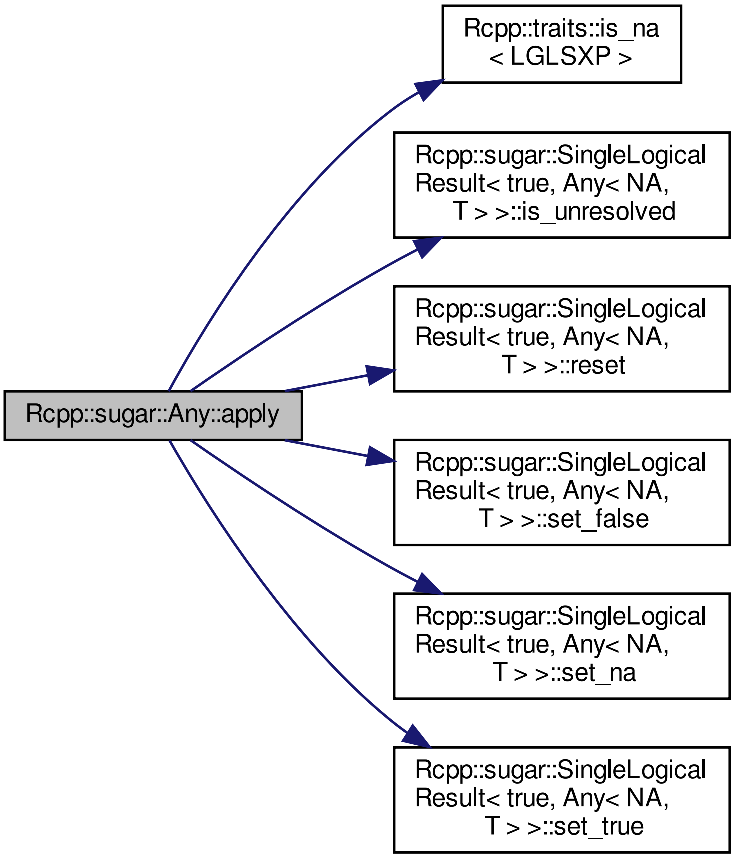
6.32.4.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::Any< NA, T >::apply ( )  [inline]
```

Definition at line 35 of file any.h.

References Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::is_unresolved(), Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::reset(), Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_false(), Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_na(), and Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_true().

Here is the call graph for this function:


Rcpp::sugar::Any::apply
Rcpp::traits::is_na < LGLSXP >
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::is_unresolved
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::reset
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::set_false
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::set_na
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::set_true

6.32.5 Member Data Documentation

6.32.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Any< NA, T >::object  [private]
```

Definition at line 54 of file any.h.

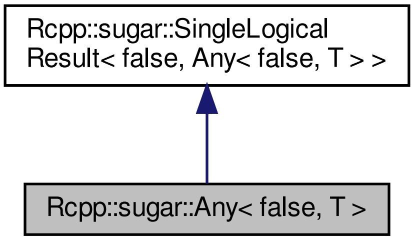
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/any.h

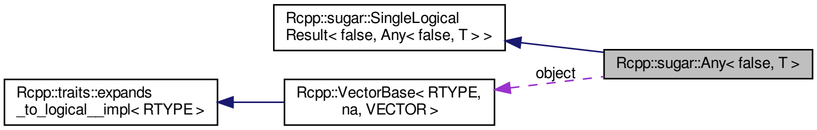
6.33 Rcpp::sugar::Any< false, T > Class Template Reference

```
#include <any.h>
```

Inheritance diagram for Rcpp::sugar::Any< false, T >:


Rcpp::sugar::SingleLogical
Result< false, Any< false, T > >
Rcpp::sugar::Any< false, T >

Collaboration diagram for Rcpp::sugar::Any< false, T >:


Rcpp::sugar::SingleLogical
Result< false, Any< false, T > >
Rcpp::sugar::Any< false, T >
object
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, T > VEC_TYPE
- typedef SingleLogicalResult< false, Any< false, T > > PARENT

Public Member Functions

- Any (const VEC_TYPE &t)
- void apply ()

Private Attributes

- const VEC_TYPE & object

Additional Inherited Members

6.33.1 Detailed Description

template<typename T>
class Rcpp::sugar::Any< false, T >

Definition at line 58 of file any.h.

6.33.2 Member Typedef Documentation

6.33.2.1 PARENT

```
template<typename T >
typedef SingleLogicalResult< false , Any<false,T> > Rcpp::sugar::Any< false, T >::PARENT
```

Definition at line 61 of file any.h.

6.33.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<LGLSXP,false,T> Rcpp::sugar::Any< false, T >::VEC_TYPE
```

Definition at line 60 of file any.h.

6.33.3 Constructor & Destructor Documentation

6.33.3.1 Any()

```
template<typename T >
Rcpp::sugar::Any< false, T >::Any (
             const VEC_TYPE & t )  [inline]
```

Definition at line 62 of file any.h.

6.33.4 Member Function Documentation

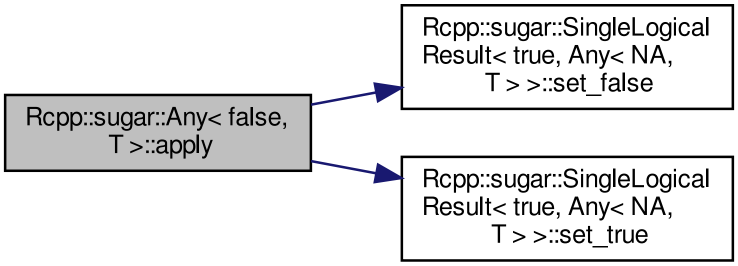
6.33.4.1 apply()

```
template<typename T >
void Rcpp::sugar::Any< false, T >::apply ( )  [inline]
```

Definition at line 64 of file any.h.

References Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_false(), and Rcpp::sugar::SingleLogical↩ Result< true, Any< NA, T > >::set_true().

Here is the call graph for this function:


Rcpp::sugar::Any< false, T >::apply
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::set_false
Rcpp::sugar::SingleLogical Result< true, Any< NA, T > >::set_true

6.33.5 Member Data Documentation

6.33.5.1 object

```
template<typename T >
const VEC_TYPE& Rcpp::sugar::Any< false, T >::object  [private]
```

Definition at line 75 of file any.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/any.h

6.34 Rcpp::Argument Class Reference

```
#include <Named.h>
```

Public Member Functions

- Argument ()
- Argument (const std::string &name_)
- template<typename T >
  traits::named_object< T > operator= (const T &t)

Public Attributes

- std::string name

6.34.1 Detailed Description

Definition at line 27 of file Named.h.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 Argument() [1/2]

```
Rcpp::Argument::Argument ( )  [inline]
```

Definition at line 29 of file Named.h.

6.34.2.2 Argument() [2/2]

```
Rcpp::Argument::Argument (
             const std::string & name_ )  [inline]
```

Definition at line 30 of file Named.h.

6.34.3 Member Function Documentation

6.34.3.1 operator=()

```
template<typename T >
traits::named_object<T> Rcpp::Argument::operator= (
             const T & t )  [inline]
```

Definition at line 33 of file Named.h.

References name.

6.34.4 Member Data Documentation

6.34.4.1 name

```
std::string Rcpp::Argument::name
```

Definition at line 37 of file Named.h.

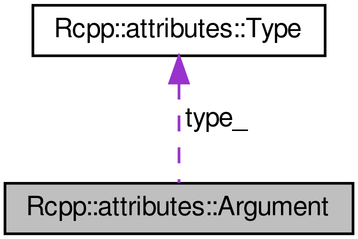
Referenced by operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Named.h

6.35 Rcpp::attributes::Argument Class Reference

Collaboration diagram for Rcpp::attributes::Argument:



Public Member Functions

- Argument ()
- Argument (const std::string &name, const Type &type, const std::string &defaultValue)
- bool empty () const
- bool operator== (const Argument &other) const
- bool operator!= (const Argument &other) const
- const std::string & name () const
- const Type & type () const
- const std::string & defaultValue () const

Private Attributes

- std::string name_
- Type type_
- std::string defaultValue_

6.35.1 Detailed Description

Definition at line 211 of file attributes.cpp.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 Argument() [1/2]

```
Rcpp::attributes::Argument::Argument ( )  [inline]
```

Definition at line 213 of file attributes.cpp.

6.35.2.2 Argument() [2/2]

```
Rcpp::attributes::Argument::Argument (
            const std::string & name,
            const Type & type,
            const std::string & defaultValue )  [inline]
```

Definition at line 214 of file attributes.cpp.

6.35.3 Member Function Documentation

6.35.3.1 defaultValue()

```
const std::string& Rcpp::attributes::Argument::defaultValue ( ) const  [inline]
```

Definition at line 236 of file attributes.cpp.

References defaultValue_.

Referenced by Rcpp::attributes::generateRArgList(), and Rcpp::attributes::printArgument().

6.35.3.2 empty()

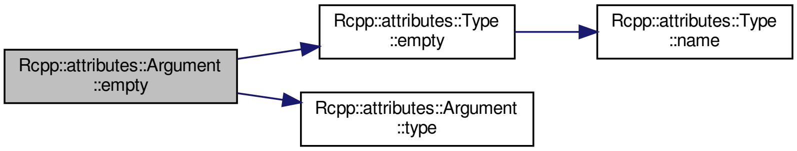
```
bool Rcpp::attributes::Argument::empty ( ) const  [inline]
```

Definition at line 221 of file attributes.cpp.

References Rcpp::attributes::Type::empty(), and type().

Referenced by Rcpp::attributes::printArgument().

Here is the call graph for this function:


Rcpp::attributes::Argument ::empty
Rcpp::attributes::Type ::empty
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::type

6.35.3.3 name()

```
const std::string& Rcpp::attributes::Argument::name ( ) const  [inline]
```

Definition at line 234 of file attributes.cpp.

References name_.

Referenced by Rcpp::attributes::checkRSignature(), Rcpp::attributes::generateCpp(), Rcpp::attributes::generateRArg↩ List(), and Rcpp::attributes::printArgument().

6.35.3.4 operator"!=()

```
bool Rcpp::attributes::Argument::operator!= (
             const Argument & other ) const  [inline]
```

Definition at line 229 of file attributes.cpp.

6.35.3.5 operator==()

```
bool Rcpp::attributes::Argument::operator== (
             const Argument & other ) const  [inline]
```

Definition at line 223 of file attributes.cpp.

References defaultValue_, name_, and type_.

6.35.3.6 type()

```
const Type& Rcpp::attributes::Argument::type ( ) const  [inline]
```

Definition at line 235 of file attributes.cpp.

References type_.

Referenced by empty(), Rcpp::attributes::generateCpp(), Rcpp::attributes::generateRArgList(), and Rcpp::attributes↩ ::printArgument().

6.35.4 Member Data Documentation

6.35.4.1 defaultValue_

```
std::string Rcpp::attributes::Argument::defaultValue_  [private]
```

Definition at line 241 of file attributes.cpp.

Referenced by defaultValue(), and operator==().

6.35.4.2 name_

```
std::string Rcpp::attributes::Argument::name_  [private]
```

Definition at line 239 of file attributes.cpp.

Referenced by name(), and operator==().

6.35.4.3 type_

```
Type Rcpp::attributes::Argument::type_  [private]
```

Definition at line 240 of file attributes.cpp.

Referenced by operator==(), and type().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.36 Rcpp::Armor< T > Class Template Reference

```
#include <Armor.h>
```

Public Member Functions

- Armor ()
- template<typename U >
  Armor (U x)
- operator SEXP () const
- template<typename U >
  Armor & operator= (const U &x)
- ∼Armor ()
- template<typename U >
  Armor< T > & operator= (const U &x)

Private Member Functions

- void init (SEXP x)
- Armor (const Armor &)
- Armor & operator= (const Armor &)

Private Attributes

- SEXP data
- PROTECT_INDEX index

6.36.1 Detailed Description

template<typename T>
class Rcpp::Armor< T >

Definition at line 24 of file Armor.h.

6.36.2 Constructor & Destructor Documentation

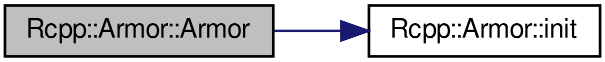
6.36.2.1 Armor() [1/3]

```
template<typename T >
Rcpp::Armor< T >::Armor ( )  [inline]
```

Definition at line 27 of file Armor.h.

References Rcpp::Armor< T >::init().

Here is the call graph for this function:



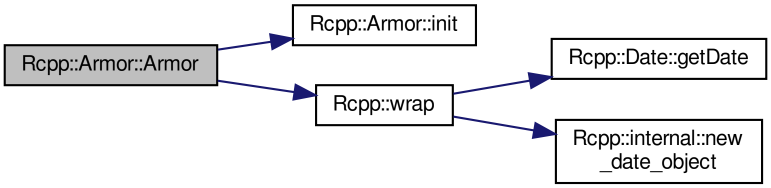
6.36.2.2 Armor() [2/3]

```
template<typename T >
template<typename U >
Rcpp::Armor< T >::Armor (
             U x )
```

Definition at line 25 of file protection.h.

References Rcpp::Armor< T >::init(), and Rcpp::wrap().

Here is the call graph for this function:



6.36.2.3 ∼Armor()

```
template<typename T >
Rcpp::Armor< T >::∼Armor ( )  [inline]
```

Definition at line 40 of file Armor.h.

6.36.2.4 Armor() [3/3]

```
template<typename T >
Rcpp::Armor< T >::Armor (
             const Armor< T > &  )  [private]
```

6.36.3 Member Function Documentation

6.36.3.1 init()

```
template<typename T >
void Rcpp::Armor< T >::init (
             SEXP x )  [inline], [private]
```

Definition at line 45 of file Armor.h.

References Rcpp::Armor< T >::data, and Rcpp::Armor< T >::index.

Referenced by Rcpp::Armor< T >::Armor().

6.36.3.2 operator SEXP()

```
template<typename T >
Rcpp::Armor< T >::operator SEXP ( ) const  [inline]
```

Definition at line 33 of file Armor.h.

References Rcpp::Armor< T >::data.

6.36.3.3 operator=() [1/3]

```
template<typename T >
Armor& Rcpp::Armor< T >::operator= (
             const Armor< T > &  )  [private]
```

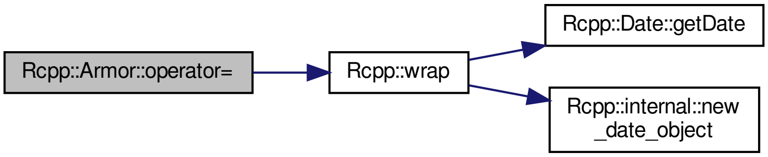
6.36.3.4 operator=() [2/3]

```
template<typename T >
template<typename U >
Armor<T>& Rcpp::Armor< T >::operator= (
             const U & x )  [inline]
```

Definition at line 31 of file protection.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.36.3.5 operator=() [3/3]

```
template<typename T >
template<typename U >
Armor& Rcpp::Armor< T >::operator= (
             const U & x )  [inline]
```

6.36.4 Member Data Documentation

6.36.4.1 data

```
template<typename T >
SEXP Rcpp::Armor< T >::data  [private]
```

Definition at line 49 of file Armor.h.

Referenced by Rcpp::Armor< T >::init(), and Rcpp::Armor< T >::operator SEXP().

6.36.4.2 index

```
template<typename T >
PROTECT_INDEX Rcpp::Armor< T >::index  [private]
```

Definition at line 50 of file Armor.h.

Referenced by Rcpp::Armor< T >::init().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/protection/Armor.h
- inst/include/Rcpp/api/meat/protection.h

6.37 Rcpp::attributes::Attribute Class Reference

Public Member Functions

- Attribute ()
- Attribute (const std::string &name, const std::vector< Param > &params, const Function &function, const std↩ ::vector< std::string > &roxygen)
- bool empty () const
- bool operator== (const Attribute &other) const
- bool operator!= (const Attribute &other) const
- const std::string & name () const
- const std::vector< Param > & params () const
- Param paramNamed (const std::string &name) const
- bool hasParameter (const std::string &name) const
- const Function & function () const
- bool isExportedFunction () const
- std::string exportedName () const
- std::string exportedCppName () const
- bool rng () const
- bool invisible () const
- const std::vector< std::string > & roxygen () const
- std::string customRSignature () const

Private Attributes

- std::string name_
- std::vector< Param > params_
- Function function_
- std::vector< std::string > roxygen_

6.37.1 Detailed Description

Definition at line 314 of file attributes.cpp.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 Attribute() [1/2]

```
Rcpp::attributes::Attribute::Attribute ( )  [inline]
```

Definition at line 316 of file attributes.cpp.

6.37.2.2 Attribute() [2/2]

```
Rcpp::attributes::Attribute::Attribute (
            const std::string & name,
            const std::vector< Param > & params,
            const Function & function,
            const std::vector< std::string > & roxygen )  [inline]
```

Definition at line 317 of file attributes.cpp.

6.37.3 Member Function Documentation

6.37.3.1 customRSignature()

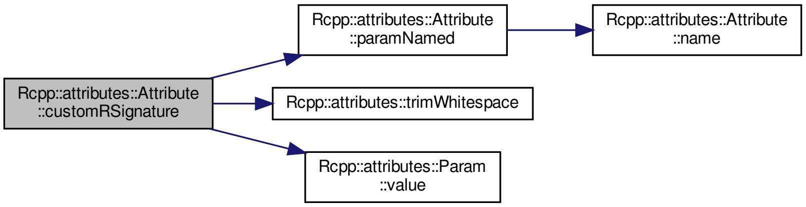
```
std::string Rcpp::attributes::Attribute::customRSignature ( ) const  [inline]
```

Definition at line 399 of file attributes.cpp.

References Rcpp::attributes::kExportSignature, paramNamed(), Rcpp::attributes::trimWhitespace(), and Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::customRSignature
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Attribute ::name
Rcpp::attributes::trimWhitespace
Rcpp::attributes::Param ::value

6.37.3.2 empty()

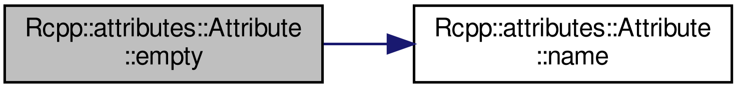
```
bool Rcpp::attributes::Attribute::empty ( ) const  [inline]
```

Definition at line 325 of file attributes.cpp.

References name().

Referenced by exportedName(), isExportedFunction(), and Rcpp::attributes::operator<<().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name

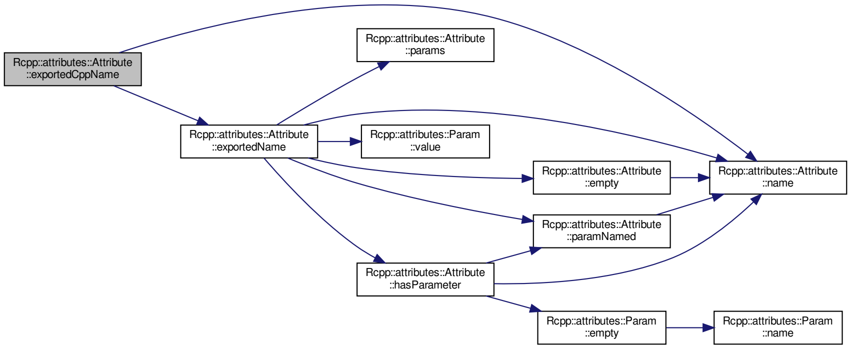
6.37.3.3 exportedCppName()

```
std::string Rcpp::attributes::Attribute::exportedCppName ( ) const  [inline]
```

Definition at line 373 of file attributes.cpp.

References exportedName(), and name().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::exportedCppName
Rcpp::attributes::Attribute ::params
Rcpp::attributes::Attribute ::exportedName
Rcpp::attributes::Param ::value
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Attribute ::hasParameter
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name

6.37.3.4 exportedName()

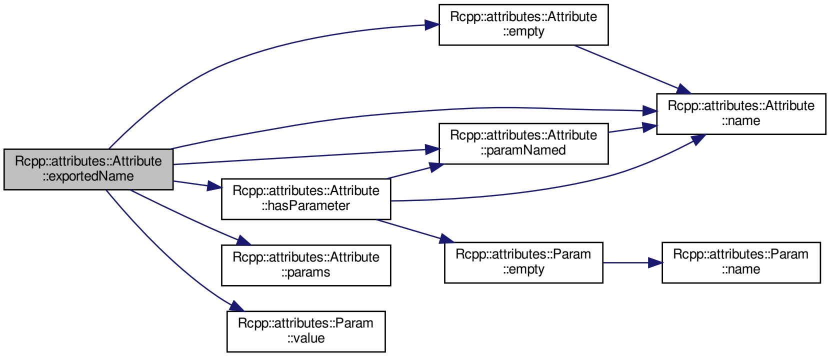
```
std::string Rcpp::attributes::Attribute::exportedName ( ) const  [inline]
```

Definition at line 355 of file attributes.cpp.

References empty(), hasParameter(), Rcpp::attributes::kExportName, name(), paramNamed(), params(), and Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), exportedCppName(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Attribute ::exportedName
Rcpp::attributes::Attribute ::hasParameter
Rcpp::attributes::Attribute ::params
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Param ::value

6.37.3.5 function()

```
const Function& Rcpp::attributes::Attribute::function ( ) const  [inline]
```

Definition at line 349 of file attributes.cpp.

References function_.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::generateCpp(), Rcpp::attributes::operator<<(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

6.37.3.6 hasParameter()

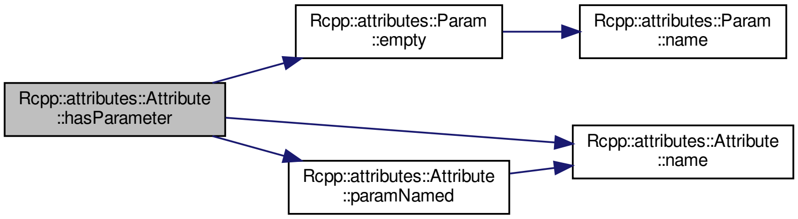
```
bool Rcpp::attributes::Attribute::hasParameter (
            const std::string & name ) const  [inline]
```

Definition at line 345 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), name(), and paramNamed().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), and exportedName().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::hasParameter
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute ::paramNamed

6.37.3.7 invisible()

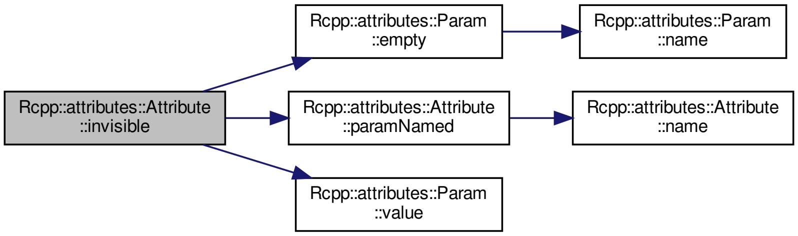
```
bool Rcpp::attributes::Attribute::invisible ( ) const  [inline]
```

Definition at line 388 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::kExportInvisible, Rcpp::attributes::kParamValueTrue, Rcpp::attributes::kParamValueTRUE, paramNamed(), and Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::invisible
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Param ::value

6.37.3.8 isExportedFunction()

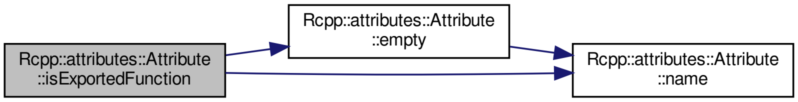
```
bool Rcpp::attributes::Attribute::isExportedFunction ( ) const  [inline]
```

Definition at line 351 of file attributes.cpp.

References empty(), Rcpp::attributes::kExportAttribute, and name().

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), and Rcpp::attributes::generateCpp().

Here is the call graph for this function:


Rcpp::attributes::Attribute ::isExportedFunction
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name

6.37.3.9 name()

```
const std::string& Rcpp::attributes::Attribute::name ( ) const  [inline]
```

Definition at line 339 of file attributes.cpp.

References name_.

Referenced by empty(), exportedCppName(), exportedName(), hasParameter(), isExportedFunction(), Rcpp::attributes::operator<<(), and paramNamed().

6.37.3.10 operator"!=()

```
bool Rcpp::attributes::Attribute::operator!= (
             const Attribute & other ) const  [inline]
```

Definition at line 334 of file attributes.cpp.

6.37.3.11 operator==()

```
bool Rcpp::attributes::Attribute::operator== (
            const Attribute & other ) const  [inline]
```

Definition at line 327 of file attributes.cpp.

References function_, name_, params_, and roxygen_.

6.37.3.12 paramNamed()

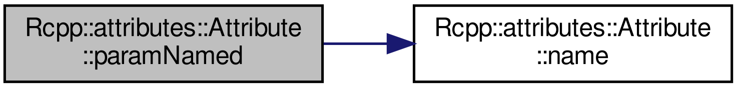
```
Param Rcpp::attributes::Attribute::paramNamed (
            const std::string & name ) const
```

Definition at line 1082 of file attributes.cpp.

References name(), and params_.

Referenced by customRSignature(), exportedName(), hasParameter(), invisible(), and rng().

Here is the call graph for this function:



6.37.3.13 params()

```
const std::vector<Param>& Rcpp::attributes::Attribute::params ( ) const  [inline]
```

Definition at line 341 of file attributes.cpp.

References params_.

Referenced by exportedName(), and Rcpp::attributes::operator<<().

6.37.3.14 rng()

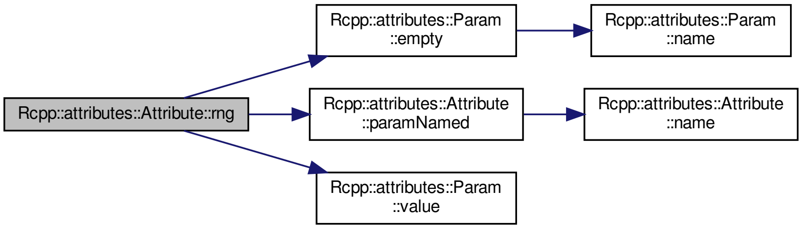
```
bool Rcpp::attributes::Attribute::rng ( ) const  [inline]
```

Definition at line 379 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::kExportRng, Rcpp::attributes::kParamValueTrue, Rcpp::attributes::kParamValueTRUE, paramNamed(), and Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::generateCpp().

Here is the call graph for this function:


Rcpp::attributes::Attribute::rng
Rcpp::attributes::Param ::empty
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Param ::value
Rcpp::attributes::Param ::name
Rcpp::attributes::Attribute ::name

6.37.3.15 roxygen()

```
const std::vector<std::string>& Rcpp::attributes::Attribute::roxygen ( ) const  [inline]
```

Definition at line 397 of file attributes.cpp.

References roxygen_.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions().

6.37.4 Member Data Documentation

6.37.4.1 function_

```
Function Rcpp::attributes::Attribute::function_  [private]
```

Definition at line 424 of file attributes.cpp.

Referenced by function(), and operator==().

6.37.4.2 name_

```
std::string Rcpp::attributes::Attribute::name_  [private]
```

Definition at line 422 of file attributes.cpp.

Referenced by name(), and operator==().

6.37.4.3 params_

```
std::vector<Param> Rcpp::attributes::Attribute::params_  [private]
```

Definition at line 423 of file attributes.cpp.

Referenced by operator==(), paramNamed(), and params().

6.37.4.4 roxygen_

```
std::vector<std::string> Rcpp::attributes::Attribute::roxygen_  [private]
```

Definition at line 425 of file attributes.cpp.

Referenced by operator==(), and roxygen().

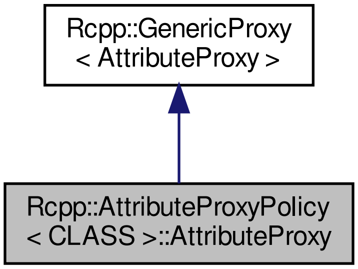
The documentation for this class was generated from the following file:

- src/attributes.cpp

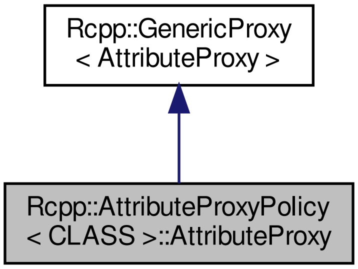
6.38 Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy Class Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy:



Collaboration diagram for Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy:



Public Member Functions

- AttributeProxy (CLASS &v, const std::string &name)
- AttributeProxy & operator= (const AttributeProxy &rhs)
- template< typename T >
  AttributeProxy & operator= (const T &rhs)
- template< typename T >
  operator T () const
- operator SEXP () const
- template< typename T >
  AttributeProxyPolicy< CLASS >::AttributeProxy & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- CLASS & parent
- SEXP attr_name

6.38.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy

Definition at line 27 of file AttributeProxy.h.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 AttributeProxy()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::AttributeProxy (
             CLASS & v,
             const std::string & name )  [inline]
```

Definition at line 29 of file AttributeProxy.h.

6.38.3 Member Function Documentation

6.38.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get ( ) const  [inline], [private]
```

Definition at line 48 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr_name, and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=().

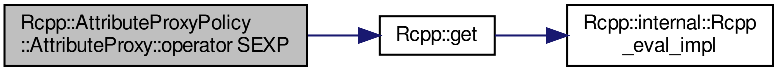
6.38.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator SEXP  [inline]
```

Definition at line 44 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.38.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator T
```

Definition at line 39 of file proxy.h.

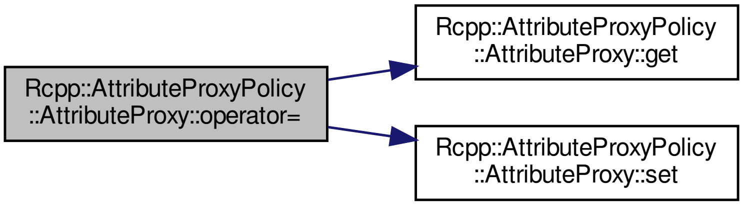
6.38.3.4 operator=() [1/3]

```
template<typename CLASS >
AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (
             const AttributeProxy & rhs )  [inline]
```

Definition at line 33 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get(), and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy
::AttributeProxy::operator=
Rcpp::AttributeProxyPolicy
::AttributeProxy::get
Rcpp::AttributeProxyPolicy
::AttributeProxy::set

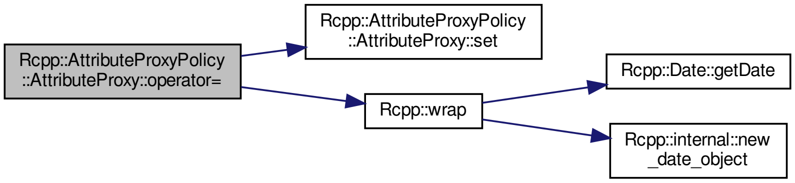
6.38.3.5 operator=() [2/3]

```
template<typename CLASS >
template<typename T >
AttributeProxyPolicy<CLASS>::AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::Attribute↩
Proxy::operator= (
             const T & rhs )
```

Definition at line 32 of file proxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set(), and Rcpp::wrap().

Here is the call graph for this function:



6.38.3.6 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (
             const T & rhs )
```

6.38.3.7 set()

```
template<typename CLASS >
void Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set (
             SEXP x )  [inline], [private]
```

Definition at line 51 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr_name, and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=().

6.38.4 Member Data Documentation

6.38.4.1 attr_name

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr_name  [private]
```

Definition at line 46 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get(), and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set().

6.38.4.2 parent

```
template<typename CLASS >
CLASS& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent  [private]
```

Definition at line 45 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get(), and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set().

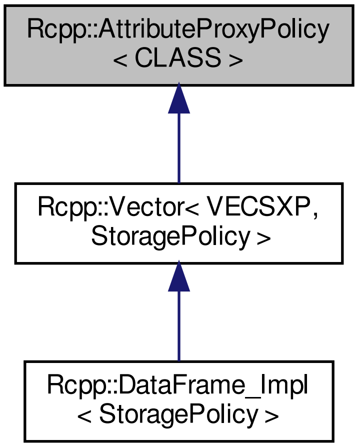
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/AttributeProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.39 Rcpp::AttributeProxyPolicy< CLASS > Class Template Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >:


Rcpp::AttributeProxyPolicy < CLASS >
Rcpp::Vector< VECSXP, StoragePolicy >
Rcpp::DataFrame_Impl < StoragePolicy >

Classes

- class AttributeProxy
- class const_AttributeProxy

Public Member Functions

- AttributeProxy attr (const std::string &name)
- const_AttributeProxy attr (const std::string &name) const
- std::vector< std::string > attributeNames () const
- bool hasAttribute (const std::string &attr) const

6.39.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >

Definition at line 24 of file AttributeProxy.h.

6.39.2 Member Function Documentation

6.39.2.1 attr() [1/2]

```
template<typename CLASS >
AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (
             const std::string & name )  [inline]
```

Definition at line 73 of file AttributeProxy.h.

Referenced by Rcpp::exception::copy_stack_trace_to_r(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_range_↩_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl(), Rcpp::DataFrame_Impl< StoragePolicy >↩::from_list(), Rcpp::AttributeProxyPolicy< CLASS >::hasAttribute(), Rcpp::Vector< RTYPE, StoragePolicy >::insert_↩_impl(), Rcpp::sugar::na_omit_impl(), Rcpp::Timer::operator SEXP(), Rcpp::newDatetimeVector::operator=(), Rcpp::↩Vector< RTYPE, StoragePolicy >::push_back__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name_↩_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push↩_front_name__impl(), Rcpp::internal::range_wrap_dispatch___impl__pair(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

6.39.2.2 attr() [2/2]

```
template<typename CLASS >
const_AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (
             const std::string & name ) const  [inline]
```

Definition at line 76 of file AttributeProxy.h.

6.39.2.3 attributeNames()

```
template<typename CLASS >
std::vector<std::string> Rcpp::AttributeProxyPolicy< CLASS >::attributeNames ( ) const  [inline]
```

Definition at line 80 of file AttributeProxy.h.

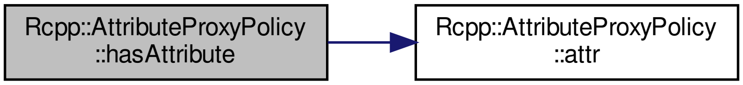
6.39.2.4 hasAttribute()

```
template<typename CLASS >
bool Rcpp::AttributeProxyPolicy< CLASS >::hasAttribute (
             const std::string & attr ) const  [inline]
```

Definition at line 90 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr().

Here is the call graph for this function:



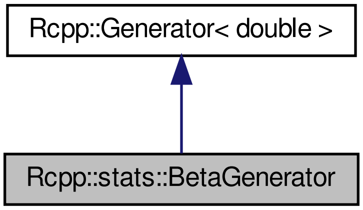
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/AttributeProxy.h

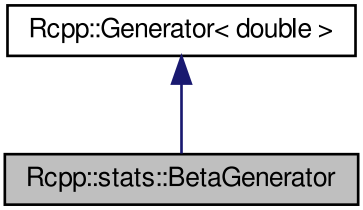
6.40 Rcpp::stats::BetaGenerator Class Reference

```
#include <rbeta.h>
```

Inheritance diagram for Rcpp::stats::BetaGenerator:



Collaboration diagram for Rcpp::stats::BetaGenerator:



Public Member Functions

- BetaGenerator (double a_, double b_)
- double operator() () const

Private Attributes

- double a
- double b

Additional Inherited Members

6.40.1 Detailed Description

Definition at line 28 of file rbeta.h.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 BetaGenerator()

```
Rcpp::stats::BetaGenerator::BetaGenerator (
             double a_,
             double b_ )  [inline]
```

Definition at line 30 of file rbeta.h.

6.40.3 Member Function Documentation

6.40.3.1 operator()()

```
double Rcpp::stats::BetaGenerator::operator() ( ) const  [inline]
```

Definition at line 32 of file rbeta.h.

References a, and b.

6.40.4 Member Data Documentation

6.40.4.1 a

```
double Rcpp::stats::BetaGenerator::a  [private]
```

Definition at line 36 of file rbeta.h.

Referenced by operator()().

6.40.4.2 b

```
double Rcpp::stats::BetaGenerator::b  [private]
```

Definition at line 36 of file rbeta.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rbeta.h

6.41 Rcpp::traits::is_convertible< T, U >::Big Struct Reference

Public Attributes

- char dummy [2]

6.41.1 Detailed Description

template<typename T, typename U>
struct Rcpp::traits::is_convertible< T, U >::Big

Definition at line 32 of file is_convertible.h.

6.41.2 Member Data Documentation

6.41.2.1 dummy

```
template<typename T , typename U >
char Rcpp::traits::is_convertible< T, U >::Big::dummy[2]
```

Definition at line 32 of file is_convertible.h.

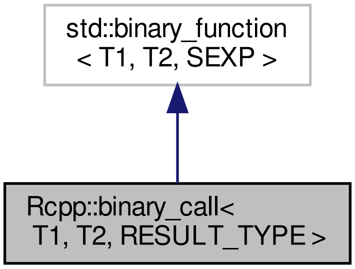
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_convertible.h

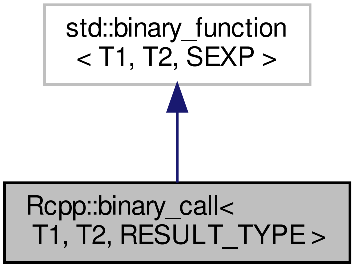
6.42 Rcpp::binary_call< T1, T2, RESULT_TYPE > Class Template Reference

```
#include <Language.h>
```

Inheritance diagram for Rcpp::binary_call< T1, T2, RESULT_TYPE >:



Collaboration diagram for Rcpp::binary_call< T1, T2, RESULT_TYPE >:



Public Member Functions

- binary_call (Language call_)
- binary_call (Language call_, R_xlen_t index1, R_xlen_t index2)
- binary_call (Function fun)
- RESULT_TYPE operator() (const T1 &o1, const T2 &o2)

Private Attributes

- Language call
- Language::Proxy proxy1
- Language::Proxy proxy2

6.42.1 Detailed Description

template<typename T1, typename T2, typename RESULT_TYPE = SEXP>
class Rcpp::binary_call< T1, T2, RESULT_TYPE >

Definition at line 201 of file Language.h.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 binary_call() [1/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
            Language call_ )  [inline]
```

Definition at line 206 of file Language.h.

6.42.2.2 binary_call() [2/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
            Language call_,
            R_xlen_t index1,
            R_xlen_t index2 )  [inline]
```

Definition at line 207 of file Language.h.

6.42.2.3 binary_call() [3/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
            Function fun )  [inline]
```

Definition at line 208 of file Language.h.

6.42.3 Member Function Documentation

6.42.3.1 operator()()

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
RESULT_TYPE Rcpp::binary_call< T1, T2, RESULT_TYPE >::operator() (
            const T1 & o1,
            const T2 & o2 )  [inline]
```

Definition at line 210 of file Language.h.

References Rcpp::unary_call< T, RESULT_TYPE >::call.

6.42.4 Member Data Documentation

6.42.4.1 call

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Language Rcpp::binary_call< T1, T2, RESULT_TYPE >::call  [private]
```

Definition at line 217 of file Language.h.

6.42.4.2 proxy1

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy1  [private]
```

Definition at line 218 of file Language.h.

6.42.4.3 proxy2

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy2  [private]
```

Definition at line 219 of file Language.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.43 Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E > Class Template Reference

```
#include <cbind.h>
```

Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type

Public Member Functions

- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- operator E& ()
- operator const E & () const

6.43.1 Detailed Description

template<int RTYPE, typename E>
class Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >

Definition at line 52 of file cbind.h.

6.43.2 Member Typedef Documentation

6.43.2.1 stored_type

```
template<int RTYPE, typename E >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E
>::stored_type
```

Definition at line 54 of file cbind.h.

6.43.3 Member Function Documentation

6.43.3.1 ncol()

```
template<int RTYPE, typename E >
R_xlen_t Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::ncol ( ) const  [inline]
```

Definition at line 72 of file cbind.h.

6.43.3.2 nrow()

```
template<int RTYPE, typename E >
R_xlen_t Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::nrow ( ) const  [inline]
```

Definition at line 68 of file cbind.h.

6.43.3.3 operator const E &()

```
template<int RTYPE, typename E >
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator const E & ( ) const  [inline]
```

Definition at line 77 of file cbind.h.

6.43.3.4 operator E&()

```
template<int RTYPE, typename E >
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator E& ( )  [inline]
```

Definition at line 76 of file cbind.h.

6.43.3.5 operator()()

```
template<int RTYPE, typename E >
stored_type Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator() (
             R_xlen_t i,
             R_xlen_t j ) const  [inline]
```

Definition at line 60 of file cbind.h.

6.43.3.6 operator[]()

```
template<int RTYPE, typename E >
stored_type Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 56 of file cbind.h.

6.43.3.7 size()

```
template<int RTYPE, typename E >
R_xlen_t Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::size ( ) const  [inline]
```

Definition at line 64 of file cbind.h.

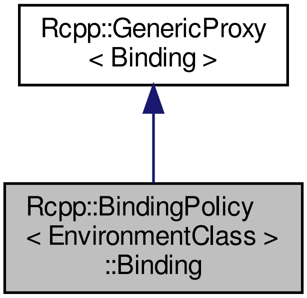
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

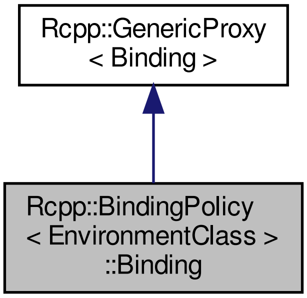
6.44 Rcpp::BindingPolicy< EnvironmentClass >::Binding Class Reference

```
#include <Binding.h>
```

Inheritance diagram for Rcpp::BindingPolicy< EnvironmentClass >::Binding:



Collaboration diagram for Rcpp::BindingPolicy< EnvironmentClass >::Binding:



Public Member Functions

- Binding (EnvironmentClass &env_, const std::string &name_)
- bool active () const
- bool locked () const
- bool exists () const
- void lock ()
- void unlock ()
- Binding & operator= (const Binding &rhs)
- template<typename WRAPPABLE >
  Binding & operator= (const WRAPPABLE &rhs)
- template<typename T >
  operator T () const
- template<typename T >
  BindingPolicy< CLASS >::Binding & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- EnvironmentClass & env
- std::string name

6.44.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::Binding

Definition at line 27 of file Binding.h.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 Binding()

```
template<typename EnvironmentClass >
Rcpp::BindingPolicy< EnvironmentClass >::Binding::Binding (
            EnvironmentClass & env_,
            const std::string & name_ )  [inline]
```

Definition at line 29 of file Binding.h.

6.44.3 Member Function Documentation

6.44.3.1 active()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::active ( ) const  [inline]
```

Definition at line 32 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.44.3.2 exists()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists ( ) const  [inline]
```

Definition at line 38 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.44.3.3 get()

```
template<typename EnvironmentClass >
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=().

6.44.3.4 lock()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock ( )  [inline]
```

Definition at line 41 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.44.3.5 locked()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked ( ) const  [inline]
```

Definition at line 35 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

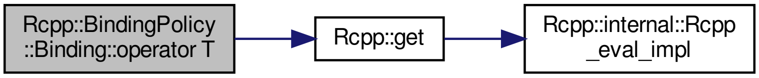
6.44.3.6 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::BindingPolicy< CLASS >::Binding::operator T
```

Definition at line 137 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:


Rcpp::BindingPolicy ::Binding::operator T
Rcpp::get
Rcpp::internal::Rcpp _eval_impl

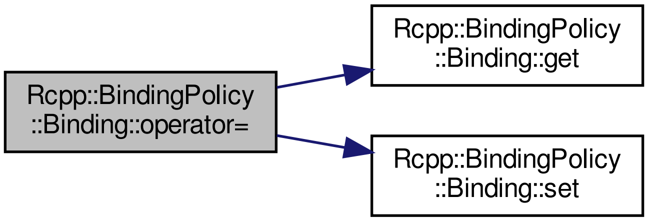
6.44.3.7 operator=() [1/3]

```
template<typename EnvironmentClass >
Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
             const Binding & rhs )  [inline]
```

Definition at line 47 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::set().

Here is the call graph for this function:



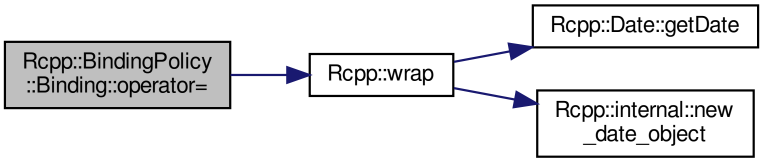
6.44.3.8 operator=() [2/3]

```
template<typename EnvironmentClass >
template<typename T >
BindingPolicy<CLASS>::Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
             const T & rhs )
```

Definition at line 130 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.44.3.9 operator=() [3/3]

```
template<typename EnvironmentClass >
template<typename WRAPPABLE >
Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
             const WRAPPABLE & rhs )
```

6.44.3.10 set()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::set (
             SEXP x )  [inline], [private]
```

Definition at line 64 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator=().

6.44.3.11 unlock()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock ( )  [inline]
```

Definition at line 44 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

6.44.4 Member Data Documentation

6.44.4.1 env

```
template<typename EnvironmentClass >
EnvironmentClass& Rcpp::BindingPolicy< EnvironmentClass >::Binding::env  [private]
```

Definition at line 68 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock().

6.44.4.2 name

```
template<typename EnvironmentClass >
std::string Rcpp::BindingPolicy< EnvironmentClass >::Binding::name  [private]
```

Definition at line 69 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::BindingPolicy< Environment↩ Class >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/Binding.h
- inst/include/Rcpp/api/meat/proxy.h

6.45 Rcpp::BindingPolicy< EnvironmentClass > Class Template Reference

```
#include <Binding.h>
```

Classes

- class Binding
- class const_Binding

Public Member Functions

- const_Binding operator[ ] (const std::string &name) const
- Binding operator[ ] (const std::string &name)

6.45.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >

Definition at line 24 of file Binding.h.

6.45.2 Member Function Documentation

6.45.2.1 operator[]() [1/2]

```
template<typename EnvironmentClass >
Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[] (
             const std::string & name )  [inline]
```

Definition at line 101 of file Binding.h.

6.45.2.2 operator[]() [2/2]

```
template<typename EnvironmentClass >
const_Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[] (
             const std::string & name ) const  [inline]
```

Definition at line 98 of file Binding.h.

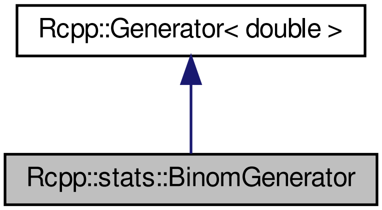
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/Binding.h

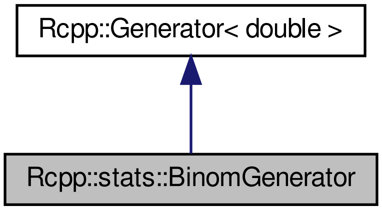
6.46 Rcpp::stats::BinomGenerator Class Reference

```
#include <rbinom.h>
```

Inheritance diagram for Rcpp::stats::BinomGenerator:



Collaboration diagram for Rcpp::stats::BinomGenerator:



Public Member Functions

- BinomGenerator (double nin_, double pp_)
- double operator() () const

Private Attributes

- double nin
- double pp

Additional Inherited Members

6.46.1 Detailed Description

Definition at line 28 of file rbinom.h.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 BinomGenerator()

```
Rcpp::stats::BinomGenerator::BinomGenerator (
            double nin_,
            double pp_ )  [inline]
```

Definition at line 30 of file rbinom.h.

6.46.3 Member Function Documentation

6.46.3.1 operator()()

```
double Rcpp::stats::BinomGenerator::operator() ( ) const  [inline]
```

Definition at line 31 of file rbinom.h.

References nin, and pp.

6.46.4 Member Data Documentation

6.46.4.1 nin

```
double Rcpp::stats::BinomGenerator::nin  [private]
```

Definition at line 35 of file rbinom.h.

Referenced by operator()().

6.46.4.2 pp

```
double Rcpp::stats::BinomGenerator::pp  [private]
```

Definition at line 35 of file rbinom.h.

Referenced by operator()().

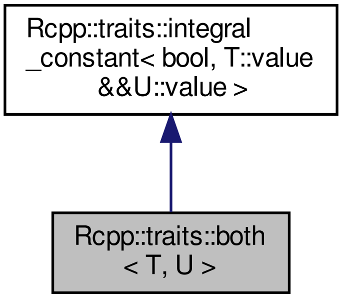
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rbinom.h

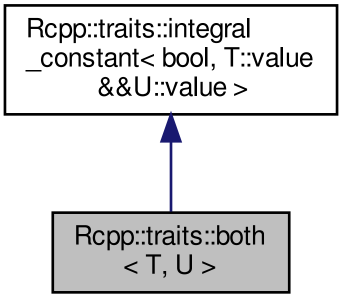
6.47 Rcpp::traits::both< T, U > Struct Template Reference

```
#include <integral_constant.h>
```

Inheritance diagram for Rcpp::traits::both< T, U >:



Collaboration diagram for Rcpp::traits::both< T, U >:



Additional Inherited Members

6.47.1 Detailed Description

template<typename T, typename U>
struct Rcpp::traits::both< T, U >

Definition at line 37 of file integral_constant.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/integral_constant.h

6.48 Cache Class Reference

```
#include <convolve10_cpp.h>
```

Public Types

- typedef double & proxy
- typedef double ∗ iterator
- typedef double & proxy
- typedef double ∗ iterator

Public Member Functions

- Cache (iterator data_)
- proxy ref (int i)
- proxy ref (int i) const
- Cache (iterator data_)
- proxy ref (int i)
- proxy ref (int i) const

Private Attributes

- iterator data

6.48.1 Detailed Description

Definition at line 2 of file convolve10_cpp.h.

6.48.2 Member Typedef Documentation

6.48.2.1 iterator [1/2]

```
typedef double* Cache::iterator
```

Definition at line 5 of file convolve10_cpp.h.

6.48.2.2 iterator [2/2]

```
typedef double* Cache::iterator
```

Definition at line 18 of file convolve9_cpp.cpp.

6.48.2.3 proxy [1/2]

```
typedef double& Cache::proxy
```

Definition at line 4 of file convolve10_cpp.h.

6.48.2.4 proxy [2/2]

```
typedef double& Cache::proxy
```

Definition at line 17 of file convolve9_cpp.cpp.

6.48.3 Constructor & Destructor Documentation

6.48.3.1 Cache() [1/2]

```
Cache::Cache (
             iterator data_ )  [inline]
```

Definition at line 7 of file convolve10_cpp.h.

6.48.3.2 Cache() [2/2]

```
Cache::Cache (
             iterator data_ )  [inline]
```

Definition at line 20 of file convolve9_cpp.cpp.

6.48.4 Member Function Documentation

6.48.4.1 ref() [1/4]

```
proxy Cache::ref (
             int i )  [inline]
```

Definition at line 9 of file convolve10_cpp.h.

References data.

Referenced by Vec::operator[ ]().

6.48.4.2 ref() [2/4]

```
proxy Cache::ref (
             int i )  [inline]
```

Definition at line 22 of file convolve9_cpp.cpp.

References data.

6.48.4.3 ref() [3/4]

```
proxy Cache::ref (
             int i ) const  [inline]
```

Definition at line 10 of file convolve10_cpp.h.

References data.

6.48.4.4 ref() [4/4]

```
proxy Cache::ref (
             int i ) const  [inline]
```

Definition at line 23 of file convolve9_cpp.cpp.

References data.

6.48.5 Member Data Documentation

6.48.5.1 data

```
iterator Cache::data  [private]
```

Definition at line 13 of file convolve10_cpp.h.

Referenced by ref().

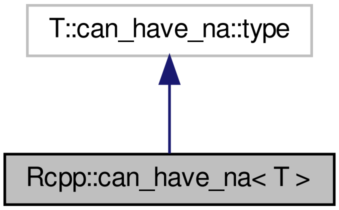
The documentation for this class was generated from the following files:

- inst/examples/ConvolveBenchmarks/convolve10_cpp.h
- inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp

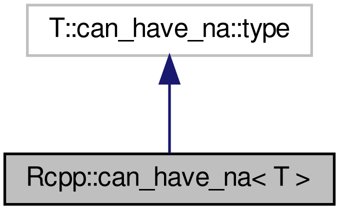
6.49 Rcpp::can_have_na< T > Struct Template Reference

```
#include <can_have_na.h>
```

Inheritance diagram for Rcpp::can_have_na< T >:



Collaboration diagram for Rcpp::can_have_na< T >:



6.49.1 Detailed Description

template<typename T>
struct Rcpp::can_have_na< T >

Definition at line 28 of file can_have_na.h.

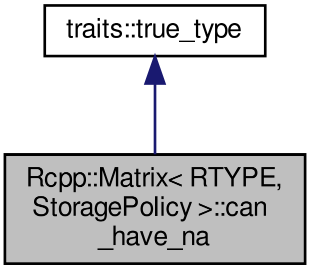
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/can_have_na.h

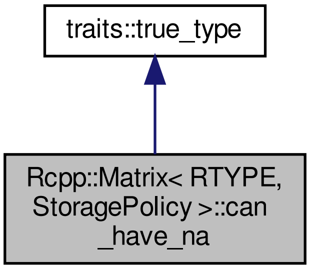
6.50 Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na Struct Reference

```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na:



Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na:



Additional Inherited Members

6.50.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na

Definition at line 35 of file Matrix.h.

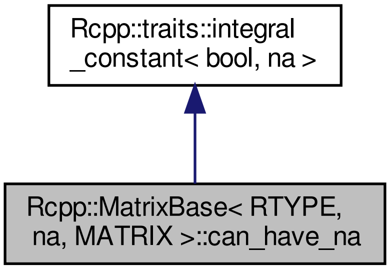
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/Matrix.h

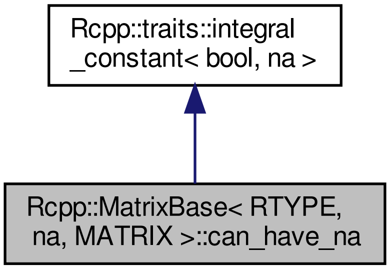
6.51 Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na Struct Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na:



Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na:


Rcpp::traits::integral_constant< bool, na >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na

Additional Inherited Members

6.51.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na

Definition at line 36 of file MatrixBase.h.

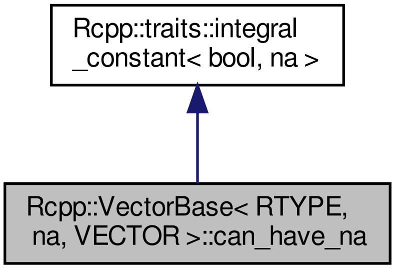
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

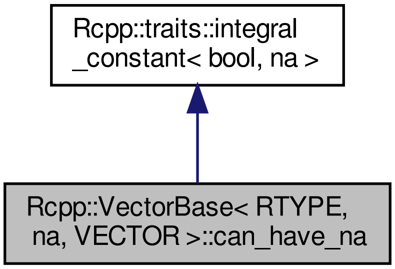
6.52 Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na Struct Reference

```
#include <VectorBase.h>
```

Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na:


Rcpp::traits::integral _constant< bool, na >
Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na

Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na:


Rcpp::traits::integral _constant< bool, na >
Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na

Additional Inherited Members

6.52.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na

Definition at line 33 of file VectorBase.h.

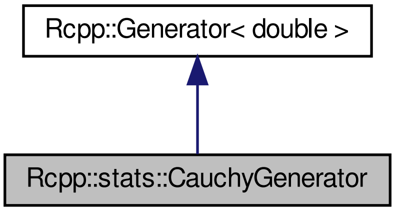
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

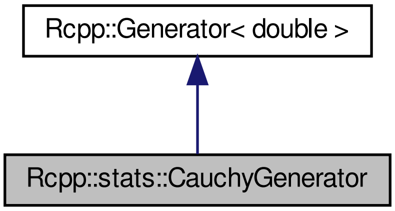
6.53 Rcpp::stats::CauchyGenerator Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for Rcpp::stats::CauchyGenerator:



Collaboration diagram for Rcpp::stats::CauchyGenerator:



Public Member Functions

- CauchyGenerator (double location_, double scale_)
- double operator() () const

Private Attributes

- double location
- double scale

Additional Inherited Members

6.53.1 Detailed Description

Definition at line 28 of file rcauchy.h.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 CauchyGenerator()

```
Rcpp::stats::CauchyGenerator::CauchyGenerator (
            double location_,
            double scale_ )  [inline]
```

Definition at line 31 of file rcauchy.h.

6.53.3 Member Function Documentation

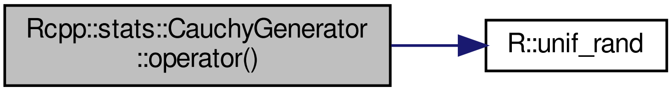
6.53.3.1 operator()()

```
double Rcpp::stats::CauchyGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rcauchy.h.

References location, scale, and R::unif_rand().

Here is the call graph for this function:



6.53.4 Member Data Documentation

6.53.4.1 location

```
double Rcpp::stats::CauchyGenerator::location  [private]
```

Definition at line 39 of file rcauchy.h.

Referenced by operator()().

6.53.4.2 scale

```
double Rcpp::stats::CauchyGenerator::scale  [private]
```

Definition at line 39 of file rcauchy.h.

Referenced by operator()().

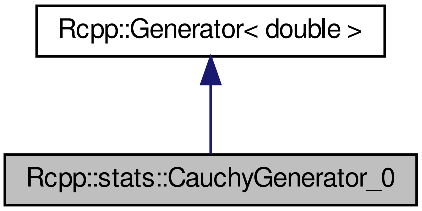
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

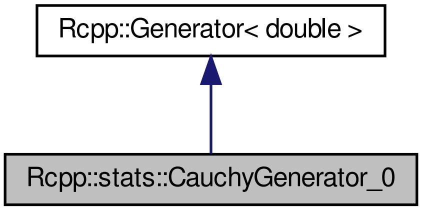
6.54 Rcpp::stats::CauchyGenerator_0 Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for Rcpp::stats::CauchyGenerator_0:



Collaboration diagram for Rcpp::stats::CauchyGenerator_0:



Public Member Functions

- CauchyGenerator_0 ()
- double operator() () const

Additional Inherited Members

6.54.1 Detailed Description

Definition at line 56 of file rcauchy.h.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 CauchyGenerator_0()

```
Rcpp::stats::CauchyGenerator_0::CauchyGenerator_0 ( )  [inline]
```

Definition at line 59 of file rcauchy.h.

6.54.3 Member Function Documentation

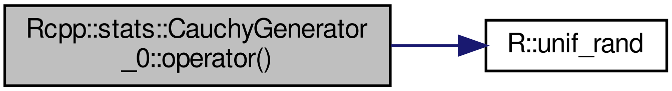
6.54.3.1 operator()()

```
double Rcpp::stats::CauchyGenerator_0::operator() ( ) const  [inline]
```

Definition at line 61 of file rcauchy.h.

References R::unif_rand().

Here is the call graph for this function:



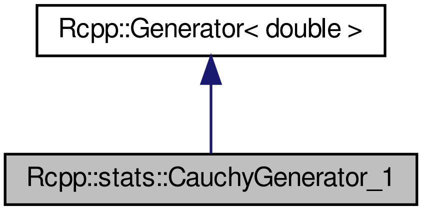
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

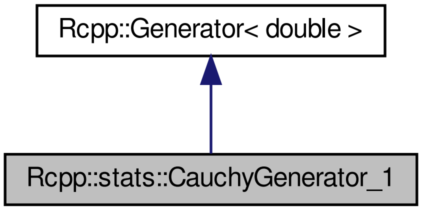
6.55 Rcpp::stats::CauchyGenerator_1 Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for Rcpp::stats::CauchyGenerator_1:



Collaboration diagram for Rcpp::stats::CauchyGenerator_1:



Public Member Functions

- CauchyGenerator_1 (double location_)
- double operator() () const

Private Attributes

- double location

Additional Inherited Members

6.55.1 Detailed Description

Definition at line 42 of file rcauchy.h.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 CauchyGenerator_1()

```
Rcpp::stats::CauchyGenerator_1::CauchyGenerator_1 (
             double location_ )  [inline]
```

Definition at line 45 of file rcauchy.h.

6.55.3 Member Function Documentation

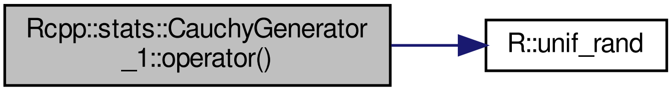
6.55.3.1 operator()()

```
double Rcpp::stats::CauchyGenerator_1::operator() ( ) const  [inline]
```

Definition at line 48 of file rcauchy.h.

References location, and R::unif_rand().

Here is the call graph for this function:



6.55.4 Member Data Documentation

6.55.4.1 location

```
double Rcpp::stats::CauchyGenerator_1::location  [private]
```

Definition at line 53 of file rcauchy.h.

Referenced by operator()().

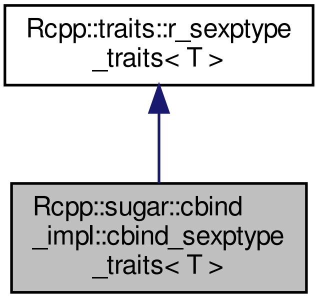
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rcauchy.h

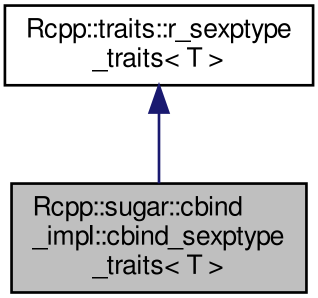
6.56 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >:


Rcpp::traits::r_sexptype _traits< T >
Rcpp::sugar::cbind _impl::cbind_sexptype _traits< T >

Collaboration diagram for Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >:


Rcpp::traits::r_sexptype _traits< T >
Rcpp::sugar::cbind _impl::cbind_sexptype _traits< T >

Additional Inherited Members

6.56.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >

Definition at line 31 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.57 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP > Struct Reference

```
#include <cbind.h>
```

Public Types

- enum { rtype = STRSXP }

6.57.1 Detailed Description

Definition at line 35 of file cbind.h.

6.57.2 Member Enumeration Documentation

6.57.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 36 of file cbind.h.

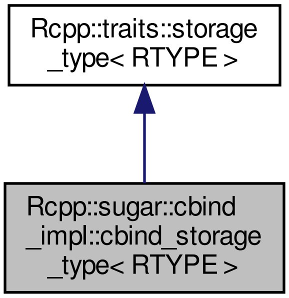
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

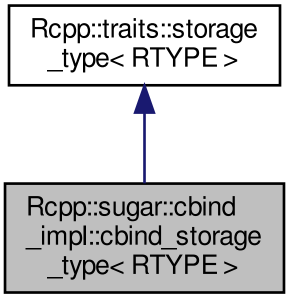
6.58 Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >:



Collaboration diagram for Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >:



Additional Inherited Members

6.58.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >

Definition at line 41 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.59 Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP > Struct Reference

```
#include <cbind.h>
```

Public Types

- typedef bool type

6.59.1 Detailed Description

Definition at line 45 of file cbind.h.

6.59.2 Member Typedef Documentation

6.59.2.1 type

```
typedef bool Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP >::type
```

Definition at line 46 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.60 Rcpp::traits::char_type< T > Struct Template Reference

```
#include <char_type.h>
```

Public Types

- typedef T::value_type type

6.60.1 Detailed Description

template<typename T>
struct Rcpp::traits::char_type< T >

Definition at line 30 of file char_type.h.

6.60.2 Member Typedef Documentation

6.60.2.1 type

```
template<typename T >
typedef T::value_type Rcpp::traits::char_type< T >::type
```

Definition at line 31 of file char_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/char_type.h

6.61 Rcpp::traits::char_type< const char * > Struct Reference

```
#include <char_type.h>
```

Public Types

- typedef char type

6.61.1 Detailed Description

Definition at line 37 of file char_type.h.

6.61.2 Member Typedef Documentation

6.61.2.1 type

```
typedef char Rcpp::traits::char_type< const char * >::type
```

Definition at line 38 of file char_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/char_type.h

6.62 Rcpp::traits::char_type< const wchar_t * > Struct Reference

```
#include <char_type.h>
```

Public Types

- typedef wchar_t type

6.62.1 Detailed Description

Definition at line 34 of file char_type.h.

6.62.2 Member Typedef Documentation

6.62.2.1 type

```
typedef wchar_t Rcpp::traits::char_type< const wchar_t * >::type
```

Definition at line 35 of file char_type.h.

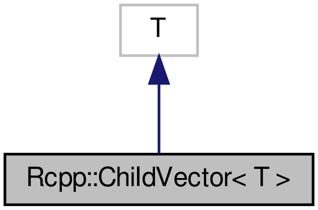
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/char_type.h

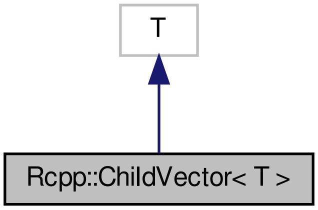
6.63 Rcpp::ChildVector< T > Class Template Reference

```
#include <ChildVector.h>
```

Inheritance diagram for Rcpp::ChildVector< T >:



Collaboration diagram for Rcpp::ChildVector< T >:



Public Member Functions

- ChildVector (SEXP data_, SEXP parent_, R_xlen_t i_)
- ChildVector (const ChildVector &other)
- ChildVector & operator= (const ChildVector &other)
- ChildVector & operator= (const T &other)
- template<typename U >
  ChildVector & operator= (const U &other)

Private Attributes

- SEXP parent
- R_xlen_t i

6.63.1 Detailed Description

template<typename T>
class Rcpp::ChildVector< T >

Definition at line 26 of file ChildVector.h.

6.63.2 Constructor & Destructor Documentation

6.63.2.1 ChildVector() [1/2]

```
template<typename T >
Rcpp::ChildVector< T >::ChildVector (
            SEXP data_,
            SEXP parent_,
            R_xlen_t i_ )  [inline]
```

Definition at line 30 of file ChildVector.h.

6.63.2.2 ChildVector() [2/2]

```
template<typename T >
Rcpp::ChildVector< T >::ChildVector (
            const ChildVector< T > & other )  [inline]
```

Definition at line 35 of file ChildVector.h.

6.63.3 Member Function Documentation

6.63.3.1 operator=() [1/3]

```
template<typename T >
ChildVector& Rcpp::ChildVector< T >::operator= (
            const ChildVector< T > & other )  [inline]
```

Definition at line 40 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, and Rcpp::ChildVector< T >::parent.

6.63.3.2 operator=() [2/3]

```
template<typename T >
ChildVector& Rcpp::ChildVector< T >::operator= (
             const T & other )  [inline]
```

Definition at line 50 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, and Rcpp::ChildVector< T >::parent.

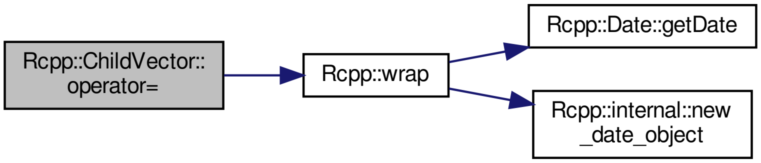
6.63.3.3 operator=() [3/3]

```
template<typename T >
template<typename U >
ChildVector& Rcpp::ChildVector< T >::operator= (
             const U & other )  [inline]
```

Definition at line 59 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, Rcpp::ChildVector< T >::parent, and Rcpp::wrap().

Here is the call graph for this function:



6.63.4 Member Data Documentation

6.63.4.1 i

```
template<typename T >
R_xlen_t Rcpp::ChildVector< T >::i  [private]
```

Definition at line 71 of file ChildVector.h.

Referenced by Rcpp::ChildVector< T >::operator=().

6.63.4.2 parent

```
template<typename T >
SEXP Rcpp::ChildVector< T >::parent  [private]
```

Definition at line 70 of file ChildVector.h.

Referenced by Rcpp::ChildVector< T >::operator=().

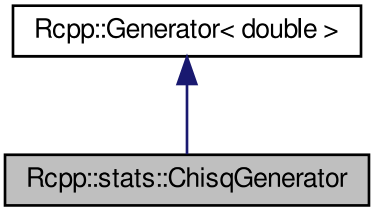
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ChildVector.h

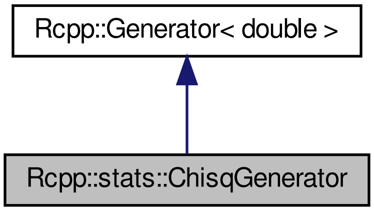
6.64 Rcpp::stats::ChisqGenerator Class Reference

```
#include <rchisq.h>
```

Inheritance diagram for Rcpp::stats::ChisqGenerator:



Collaboration diagram for Rcpp::stats::ChisqGenerator:



Public Member Functions

- ChisqGenerator (double df_)
- double operator() () const

Private Attributes

- double df_2

Additional Inherited Members

6.64.1 Detailed Description

Definition at line 28 of file rchisq.h.

6.64.2 Constructor & Destructor Documentation

6.64.2.1 ChisqGenerator()

```
Rcpp::stats::ChisqGenerator::ChisqGenerator (
             double df_ )  [inline]
```

Definition at line 31 of file rchisq.h.

6.64.3 Member Function Documentation

6.64.3.1 operator()()

```
double Rcpp::stats::ChisqGenerator::operator() ( ) const  [inline]
```

Definition at line 33 of file rchisq.h.

References df_2.

6.64.4 Member Data Documentation

6.64.4.1 df_2

```
double Rcpp::stats::ChisqGenerator::df_2  [private]
```

Definition at line 38 of file rchisq.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rchisq.h

6.65 Rcpp::sugar::clamp_operator< RTYPE, NA > Struct Template Reference

```
#include <clamp.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- clamp_operator (STORAGE lhs_, STORAGE rhs_)
- STORAGE operator() (STORAGE x) const

Public Attributes

- STORAGE lhs
- STORAGE rhs

6.65.1 Detailed Description

template<int RTYPE, bool NA>
struct Rcpp::sugar::clamp_operator< RTYPE, NA >

Definition at line 29 of file clamp.h.

6.65.2 Member Typedef Documentation

6.65.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::clamp_operator< RTYPE, NA >::STORAGE
```

Definition at line 30 of file clamp.h.

6.65.3 Constructor & Destructor Documentation

6.65.3.1 clamp_operator()

```
template<int RTYPE, bool NA>
Rcpp::sugar::clamp_operator< RTYPE, NA >::clamp_operator (
            STORAGE lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 32 of file clamp.h.

6.65.4 Member Function Documentation

6.65.4.1 operator()()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::operator() (
            STORAGE x ) const  [inline]
```

Definition at line 34 of file clamp.h.

References Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs, and Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs.

6.65.5 Member Data Documentation

6.65.5.1 lhs

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs
```

Definition at line 37 of file clamp.h.

Referenced by Rcpp::sugar::clamp_operator< REALSXP, true >::operator()(), and Rcpp::sugar::clamp_operator< RTYPE, NA >::operator()().

6.65.5.2 rhs

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs
```

Definition at line 37 of file clamp.h.

Referenced by Rcpp::sugar::clamp_operator< REALSXP, true >::operator()(), and Rcpp::sugar::clamp_operator< RTYPE, NA >::operator()().

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

6.66 Rcpp::sugar::clamp_operator< REALSXP, true > Struct Reference

```
#include <clamp.h>
```

Public Member Functions

- clamp_operator (double lhs_, double rhs_)
- double operator() (double x) const

Public Attributes

- double lhs
- double rhs

6.66.1 Detailed Description

Definition at line 41 of file clamp.h.

6.66.2 Constructor & Destructor Documentation

6.66.2.1 clamp_operator()

```
Rcpp::sugar::clamp_operator< REALSXP, true >::clamp_operator (
            double lhs_,
            double rhs_ )  [inline]
```

Definition at line 42 of file clamp.h.

6.66.3 Member Function Documentation

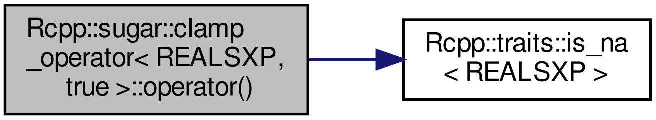
6.66.3.1 operator()()

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::operator() (
            double x ) const  [inline]
```

Definition at line 44 of file clamp.h.

References Rcpp::traits::is_na< REALSXP >(), Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs, and Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs.

Here is the call graph for this function:



6.66.4 Member Data Documentation

6.66.4.1 lhs

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::lhs
```

Definition at line 48 of file clamp.h.

6.66.4.2 rhs

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::rhs
```

Definition at line 48 of file clamp.h.

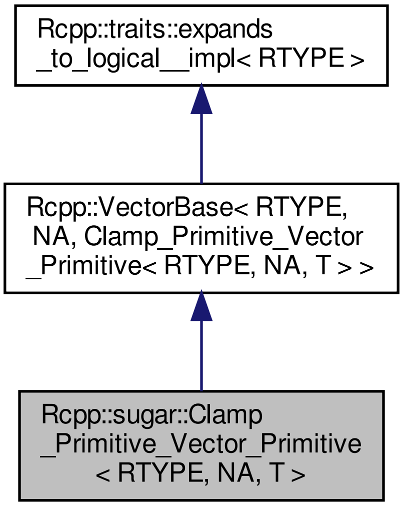
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

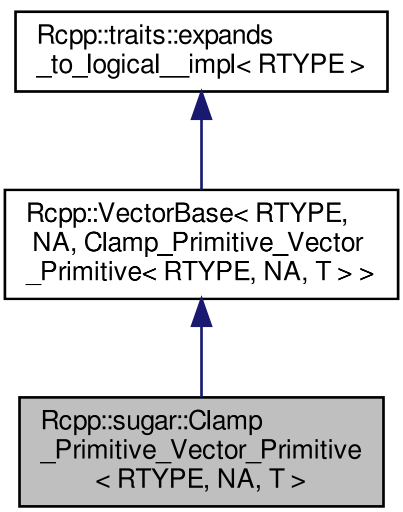
6.67 Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <clamp.h>
```

Inheritance diagram for Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
NA, Clamp_Primitive_Vector
_Primitive< RTYPE, NA, T > >
Rcpp::sugar::Clamp
_Primitive_Vector_Primitive
< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
NA, Clamp_Primitive_Vector
_Primitive< RTYPE, NA, T > >
Rcpp::sugar::Clamp
_Primitive_Vector_Primitive
< RTYPE, NA, T >

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef clamp_operator< RTYPE, NA > OPERATOR

Public Member Functions

- Clamp_Primitive_Vector_Primitive (STORAGE lhs_, const T &vec_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T & vec
- OPERATOR op

6.67.1 Detailed Description

template< int RTYPE, bool NA, typename T>
class Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >

Definition at line 57 of file clamp.h.

6.67.2 Member Typedef Documentation

6.67.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >
typedef clamp_operator<RTYPE,NA> Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >↩
::OPERATOR
```

Definition at line 64 of file clamp.h.

6.67.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Clamp_Primitive_Vector_Primitive<
RTYPE, NA, T >::STORAGE
```

Definition at line 63 of file clamp.h.

6.67.3 Constructor & Destructor Documentation

6.67.3.1 Clamp_Primitive_Vector_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::Clamp_Primitive_Vector_Primitive (
             STORAGE lhs_,
             const T & vec_,
             STORAGE rhs_ )  [inline]
```

Definition at line 66 of file clamp.h.

6.67.4 Member Function Documentation

6.67.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const [inline]
```

Definition at line 68 of file clamp.h.

References Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::op, and Rcpp::sugar::Clamp_↩ Primitive_Vector_Primitive< RTYPE, NA, T >::vec.

6.67.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 71 of file clamp.h.

References Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::vec.

6.67.5 Member Data Documentation

6.67.5.1 op

```
template<int RTYPE, bool NA, typename T >
OPERATOR Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::op [private]
```

Definition at line 75 of file clamp.h.

Referenced by Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::operator[ ]().

6.67.5.2 vec

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::vec [private]
```

Definition at line 74 of file clamp.h.

Referenced by Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::↩ Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::size().

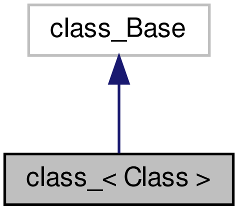
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

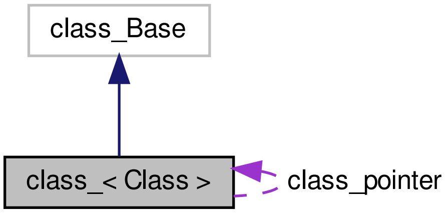
6.68 class_< Class > Class Template Reference

```
#include <class.h>
```

Inheritance diagram for class_< Class >:



Collaboration diagram for class_< Class >:



Public Types

- typedef class_< Class > self
- typedef CppMethod< Class > method_class
- typedef SignedMethod< Class > signed_method_class
- typedef std::vector< signed_method_class ∗ > vec_signed_method
- typedef std::map< std::string, vec_signed_method ∗ > map_vec_signed_method
- typedef std::pair< std::string, vec_signed_method ∗ > vec_signed_method_pair
- typedef std::map< std::string, method_class ∗ > METHOD_MAP
- typedef std::pair< const std::string, method_class ∗ > PAIR
- typedef Rcpp::XPtr< Class > XP
- typedef 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 ()

6.68.1 Detailed Description

template<typename Class>
class class_< Class >

Definition at line 26 of file class.h.

6.68.2 Member Typedef Documentation

6.68.2.1 constructor_class

```
template<typename Class >
typedef Constructor_Base<Class> class_< Class >::constructor_class
```

Definition at line 42 of file class.h.

6.68.2.2 factory_class

```
template<typename Class >
typedef Factory_Base<Class> class_< Class >::factory_class
```

Definition at line 46 of file class.h.

6.68.2.3 finalizer_class

```
template<typename Class >
typedef CppFinalizer<Class> class_< Class >::finalizer_class
```

Definition at line 40 of file class.h.

6.68.2.4 map_vec_signed_method

```
template<typename Class >
typedef std::map<std::string,vec_signed_method*> class_< Class >::map_vec_signed_method
```

Definition at line 33 of file class.h.

6.68.2.5 method_class

```
template<typename Class >
typedef CppMethod<Class> class_< Class >::method_class
```

Definition at line 29 of file class.h.

6.68.2.6 METHOD_MAP

```
template<typename Class >
typedef std::map<std::string,method_class*> class_< Class >::METHOD_MAP
```

Definition at line 36 of file class.h.

6.68.2.7 PAIR

```
template<typename Class >
typedef std::pair<const std::string,method_class*> class_< Class >::PAIR
```

Definition at line 37 of file class.h.

6.68.2.8 prop_class

```
template<typename Class >
typedef CppProperty<Class> class_< Class >::prop_class
```

Definition at line 50 of file class.h.

6.68.2.9 PROP_PAIR

```
template<typename Class >
typedef std::pair<const std::string,prop_class*> class_< Class >::PROP_PAIR
```

Definition at line 52 of file class.h.

6.68.2.10 PROPERTY_MAP

```
template<typename Class >
typedef std::map<std::string,prop_class*> class_< Class >::PROPERTY_MAP
```

Definition at line 51 of file class.h.

6.68.2.11 self

```
template<typename Class >
typedef class_<Class> class_< Class >::self
```

Definition at line 28 of file class.h.

6.68.2.12 signed_constructor_class

```
template<typename Class >
typedef SignedConstructor<Class> class_< Class >::signed_constructor_class
```

Definition at line 43 of file class.h.

6.68.2.13 signed_factory_class

```
template<typename Class >
typedef SignedFactory<Class> class_< Class >::signed_factory_class
```

Definition at line 47 of file class.h.

6.68.2.14 signed_method_class

```
template<typename Class >
typedef SignedMethod<Class> class_< Class >::signed_method_class
```

Definition at line 31 of file class.h.

6.68.2.15 vec_signed_constructor

```
template<typename Class >
typedef std::vector<signed_constructor_class*> class_< Class >::vec_signed_constructor
```

Definition at line 44 of file class.h.

6.68.2.16 vec_signed_factory

```
template<typename Class >
typedef std::vector<signed_factory_class*> class_< Class >::vec_signed_factory
```

Definition at line 48 of file class.h.

6.68.2.17 vec_signed_method

```
template<typename Class >
typedef std::vector<signed_method_class*> class_< Class >::vec_signed_method
```

Definition at line 32 of file class.h.

6.68.2.18 vec_signed_method_pair

```
template<typename Class >
typedef std::pair<std::string,vec_signed_method*> class_< Class >::vec_signed_method_pair
```

Definition at line 34 of file class.h.

6.68.2.19 XP

```
template<typename Class >
typedef Rcpp::XPtr<Class> class_< Class >::XP
```

Definition at line 39 of file class.h.

6.68.3 Constructor & Destructor Documentation

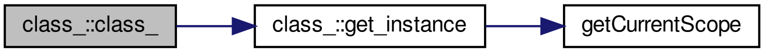
6.68.3.1 class_() [1/2]

```
template<typename Class >
class_< Class >::class_ (
            const char * name_,
            const char * doc = 0 )  [inline]
```

Definition at line 54 of file class.h.

References class_< Class >::class_pointer, and class_< Class >::get_instance().

Here is the call graph for this function:



6.68.3.2 ∼class_()

```
template<typename Class >
class_< Class >::∼class_ ( )  [inline]
```

Definition at line 98 of file class.h.

6.68.3.3 class_() [2/2]

```
template<typename Class >
class_< Class >::class_ ( )  [inline]
```

Definition at line 481 of file class.h.

6.68.4 Member Function Documentation

6.68.4.1 AddConstructor()

```
template<typename Class >
self& class_< Class >::AddConstructor (
             constructor_class * ctor,
             ValidConstructor valid,
             const char * docstring = 0 )  [inline]
```

Definition at line 100 of file class.h.

References class_< Class >::class_pointer, and class_< Class >::constructors.

6.68.4.2 AddFactory()

```
template<typename Class >
self& class_< Class >::AddFactory (
             factory_class * fact,
             ValidConstructor valid,
             const char * docstring = 0 )  [inline]
```

Definition at line 105 of file class.h.

References class_< Class >::class_pointer, and class_< Class >::factories.

6.68.4.3 AddMethod()

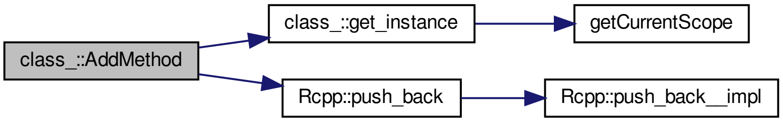
```
template<typename Class >
self& class_< Class >::AddMethod (
             const char * name_,
             method_class * m,
             ValidMethod valid = &yes,
             const char * docstring = 0 )  [inline]
```

Definition at line 239 of file class.h.

References class_< Class >::get_instance(), Rcpp::push_back(), and RCPP_DEBUG_MODULE_1.

Referenced by class_< Class >::derives().

Here is the call graph for this function:



6.68.4.4 AddProperty()

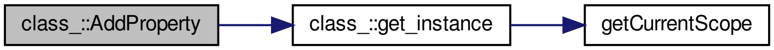
```
template<typename Class >
self& class_< Class >::AddProperty (
             const char * name_,
             prop_class * p )  [inline]
```

Definition at line 251 of file class.h.

References class_< Class >::get_instance(), and class_< Class >::properties.

Referenced by class_< Class >::derives().

Here is the call graph for this function:



6.68.4.5 complete()

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::complete ( )  [inline]
```

Definition at line 368 of file class.h.

References class_< Class >::properties, class_< Class >::specials, and class_< Class >::vec_methods.

6.68.4.6 default_constructor()

```
template<typename Class >
self& class_< Class >::default_constructor (
             const char * docstring = 0,
             ValidConstructor valid = &yes_arity<0> )  [inline]
```

Definition at line 110 of file class.h.

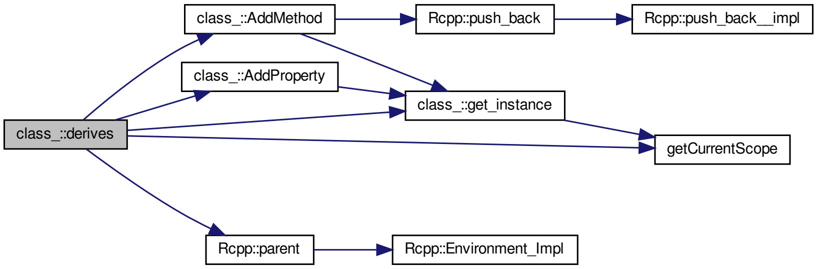
6.68.4.7 derives()

```
template<typename Class >
template<typename PARENT >
self& class_< Class >::derives (
             const char * parent )  [inline]
```

Definition at line 487 of file class.h.

References class_< Class >::AddMethod(), class_< Class >::AddProperty(), class_< Class >::get_instance(), getCurrentScope(), and Rcpp::parent().

Here is the call graph for this function:



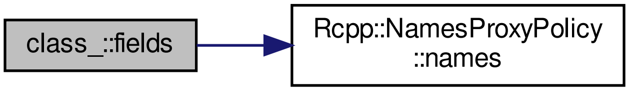
6.68.4.8 fields()

```
template<typename Class >
Rcpp::List class_< Class >::fields (
             const XP_Class & class_xp )  [inline]
```

Definition at line 409 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class_< Class >::properties.

Here is the call graph for this function:



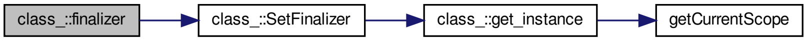
6.68.4.9 finalizer()

```
template<typename Class >
self& class_< Class >::finalizer (
             void(*)(Class *) f )  [inline]
```

Definition at line 455 of file class.h.

References class_< Class >::SetFinalizer().

Here is the call graph for this function:



6.68.4.10 get_instance()

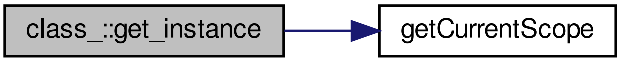
```
template<typename Class >
self* class_< Class >::get_instance ( )  [inline], [private]
```

Definition at line 70 of file class.h.

References class_< Class >::class_pointer, DEMANGLE, class_< Class >::finalizer_pointer, getCurrentScope(), RCPP_DEBUG_MODULE_2, and class_< Class >::typeinfo_name.

Referenced by class_< Class >::AddMethod(), class_< Class >::AddProperty(), class_< Class >::class_(), class_< Class >::derives(), and class_< Class >::SetFinalizer().

Here is the call graph for this function:



6.68.4.11 get_typeinfo_name()

```
template<typename Class >
std::string class_< Class >::get_typeinfo_name ( )  [inline]
```

Definition at line 119 of file class.h.

References class_< Class >::typeinfo_name.

6.68.4.12 getConstructors()

```
template<typename Class >
Rcpp::List class_< Class >::getConstructors (
              const XP_Class & class_xp,
              std::string & buffer )  [inline]
```

Definition at line 441 of file class.h.

References class_< Class >::constructors.

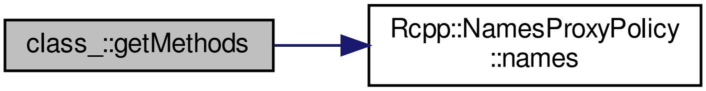
6.68.4.13 getMethods()

```
template<typename Class >
Rcpp::List class_< Class >::getMethods (
             const XP_Class & class_xp,
             std::string & buffer )  [inline]
```

Definition at line 422 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), RCPP_DEBUG_MODULE, and class_< Class >::vec_↩ methods.

Here is the call graph for this function:



6.68.4.14 getProperty()

```
template<typename Class >
SEXP class_< Class >::getProperty (
             SEXP field_xp,
             SEXP object )  [inline]
```

Definition at line 394 of file class.h.

References BEGIN_RCPP, and END_RCPP.

6.68.4.15 has_default_constructor()

```
template<typename Class >
bool class_< Class >::has_default_constructor ( )  [inline]
```

Definition at line 151 of file class.h.

References class_< Class >::constructors, and class_< Class >::factories.

6.68.4.16 has_method()

```
template<typename Class >
bool class_< Class >::has_method (
             const std::string & m )  [inline]
```

Definition at line 259 of file class.h.

References class_< Class >::vec_methods.

6.68.4.17 has_property()

```
template<typename Class >
bool class_< Class >::has_property (
             const std::string & m )  [inline]
```

Definition at line 262 of file class.h.

References class_< Class >::properties.

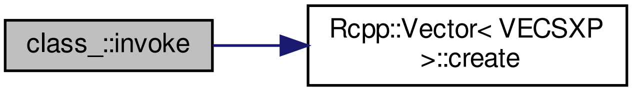
6.68.4.18 invoke()

```
template<typename Class >
SEXP class_< Class >::invoke (
             SEXP method_xp,
             SEXP object,
             SEXP * args,
             int nargs )  [inline]
```

Definition at line 167 of file class.h.

References BEGIN_RCPP, Rcpp::Vector< VECSXP >::create(), and END_RCPP.

Here is the call graph for this function:



6.68.4.19 invoke_notvoid()

```
template<typename Class >
SEXP class_< Class >::invoke_notvoid (
            SEXP method_xp,
            SEXP object,
            SEXP * args,
            int nargs ) [inline]
```

Definition at line 216 of file class.h.

References BEGIN_RCPP, and END_RCPP.

6.68.4.20 invoke_void()

```
template<typename Class >
SEXP class_< Class >::invoke_void (
            SEXP method_xp,
            SEXP object,
            SEXP * args,
            int nargs ) [inline]
```

Definition at line 194 of file class.h.

References BEGIN_RCPP, and END_RCPP.

6.68.4.21 method_names()

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::method_names ( ) [inline]
```

Definition at line 276 of file class.h.

References class_< Class >::vec_methods.

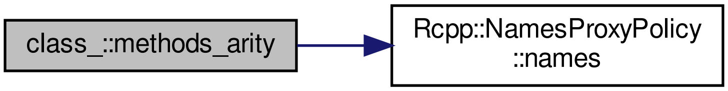
6.68.4.22 methods_arity()

```
template<typename Class >
Rcpp::IntegerVector class_< Class >::methods_arity ( )  [inline]
```

Definition at line 296 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class_< Class >::vec_methods.

Here is the call graph for this function:



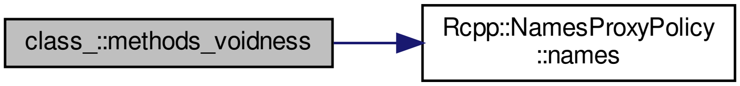
6.68.4.23 methods_voidness()

```
template<typename Class >
Rcpp::LogicalVector class_< Class >::methods_voidness ( )  [inline]
```

Definition at line 320 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class_< Class >::vec_methods.

Here is the call graph for this function:



6.68.4.24 newInstance()

```
template<typename Class >
SEXP class_< Class >::newInstance (
             SEXP * args,
             int nargs )  [inline]
```

Definition at line 123 of file class.h.

References BEGIN_RCPP, class_< Class >::constructors, END_RCPP, and class_< Class >::factories.

6.68.4.25 property_class()

```
template<typename Class >
std::string class_< Class >::property_class (
             const std::string & p )  [inline]
```

Definition at line 270 of file class.h.

References class_< Class >::properties.

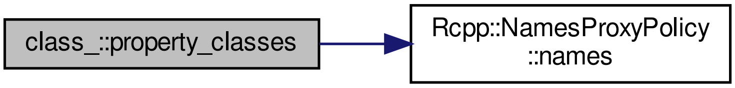
6.68.4.26 property_classes()

```
template<typename Class >
Rcpp::List class_< Class >::property_classes ( )  [inline]
```

Definition at line 355 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class_< Class >::properties.

Here is the call graph for this function:



6.68.4.27 property_is_readonly()

```
template<typename Class >
bool class_< Class >::property_is_readonly (
             const std::string & p )  [inline]
```

Definition at line 265 of file class.h.

References class_< Class >::properties.

6.68.4.28 property_names()

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::property_names ( )  [inline]
```

Definition at line 345 of file class.h.

References class_< Class >::properties.

6.68.4.29 run_finalizer()

```
template<typename Class >
virtual void class_< Class >::run_finalizer (
             SEXP object )  [inline], [virtual]
```

Definition at line 460 of file class.h.

References class_< Class >::finalizer_pointer.

6.68.4.30 SetFinalizer()

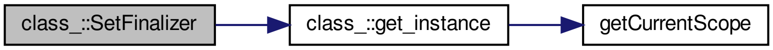
```
template<typename Class >
void class_< Class >::SetFinalizer (
             finalizer_class * f )  [inline]
```

Definition at line 464 of file class.h.

References class_< Class >::get_instance().

Referenced by class_< Class >::finalizer().

Here is the call graph for this function:



6.68.4.31 setProperty()

```
template<typename Class >
void class_< Class >::setProperty (
             SEXP field_xp,
             SEXP object,
             SEXP value ) [inline]
```

Definition at line 401 of file class.h.

References BEGIN_RCPP, and VOID_END_RCPP.

6.68.5 Member Data Documentation

6.68.5.1 class_pointer

```
template<typename Class >
self* class_< Class >::class_pointer
```

Definition at line 477 of file class.h.

Referenced by class_< Class >::AddConstructor(), class_< Class >::AddFactory(), class_< Class >::class_(), and class_< Class >::get_instance().

6.68.5.2 constructors

```
template<typename Class >
vec_signed_constructor class_< Class >::constructors
```

Definition at line 475 of file class.h.

Referenced by class_< Class >::AddConstructor(), class_< Class >::getConstructors(), class_< Class >::has_↩ default_constructor(), and class_< Class >::newInstance().

6.68.5.3 factories

```
template<typename Class >
vec_signed_factory class_< Class >::factories
```

Definition at line 476 of file class.h.

Referenced by class_< Class >::AddFactory(), class_< Class >::has_default_constructor(), and class_< Class >↩ ::newInstance().

6.68.5.4 finalizer_pointer

```
template<typename Class >
finalizer_class* class_< Class >::finalizer_pointer
```

Definition at line 473 of file class.h.

Referenced by class_< Class >::get_instance(), and class_< Class >::run_finalizer().

6.68.5.5 properties

```
template<typename Class >
PROPERTY_MAP class_< Class >::properties
```

Definition at line 471 of file class.h.

Referenced by class_< Class >::AddProperty(), class_< Class >::complete(), class_< Class >::fields(), class_↩ < Class >::has_property(), class_< Class >::property_class(), class_< Class >::property_classes(), class_< Class >::property_is_readonly(), and class_< Class >::property_names().

6.68.5.6 specials

```
template<typename Class >
int class_< Class >::specials
```

Definition at line 474 of file class.h.

Referenced by class_< Class >::complete().

6.68.5.7 typeinfo_name

```
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().

6.68.5.8 vec_methods

```
template<typename Class >
map_vec_signed_method class_< Class >::vec_methods
```

Definition at line 470 of file class.h.

Referenced by class_< Class >::complete(), class_< Class >::getMethods(), class_< Class >::has_method(), class_< Class >::method_names(), class_< Class >::methods_arity(), and class_< Class >::methods_voidness().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/class.h

6.69 Rcpp::class_Base Class Reference

```
#include <class_Base.h>
```

Public Types

- typedef Rcpp::XPtr< class_Base > XP_Class
- typedef std::map< std::string, int > ENUM
- typedef std::map< std::string, ENUM > ENUM_MAP
- typedef ENUM_MAP::value_type ENUM_MAP_PAIR

Public Member Functions

- class_Base ()
- class_Base (const char *name_, const char *doc)
- virtual Rcpp::List fields (const XP_Class &)
- virtual Rcpp::List getMethods (const XP_Class &, std::string &)
- virtual Rcpp::List getConstructors (const XP_Class &, std::string &)
- virtual void run_finalizer (SEXP)
- virtual bool has_default_constructor ()
- virtual bool has_method (const std::string &)
- virtual bool has_property (const std::string &)
- virtual SEXP newInstance (SEXP *, int)
- virtual SEXP invoke (SEXP, SEXP, SEXP *, int)
- virtual SEXP invoke_void (SEXP, SEXP, SEXP *, int)
- virtual SEXP invoke_notvoid (SEXP, SEXP, SEXP *, int)
- virtual Rcpp::CharacterVector method_names ()
- virtual Rcpp::CharacterVector property_names ()
- virtual bool property_is_readonly (const std::string &)
- virtual std::string property_class (const std::string &)
- virtual Rcpp::IntegerVector methods_arity ()
- virtual Rcpp::LogicalVector methods_voidness ()
- virtual Rcpp::List property_classes ()
- virtual Rcpp::CharacterVector complete ()
- virtual ∼class_Base ()
- virtual SEXP getProperty (SEXP, SEXP)
- virtual void setProperty (SEXP, SEXP, SEXP)
- virtual std::string get_typeinfo_name ()
- bool has_typeinfo_name (const std::string &name_)
- void add_enum (const std::string &enum_name, const std::map< std::string, int > &value)

Public Attributes

- std::string name
- std::string docstring
- ENUM_MAP enums
- std::vector< std::string > parents

6.69.1 Detailed Description

Definition at line 26 of file class_Base.h.

6.69.2 Member Typedef Documentation

6.69.2.1 ENUM

```
typedef std::map< std::string, int > Rcpp::class_Base::ENUM
```

Definition at line 88 of file class_Base.h.

6.69.2.2 ENUM_MAP

```
typedef std::map< std::string, ENUM > Rcpp::class_Base::ENUM_MAP
```

Definition at line 89 of file class_Base.h.

6.69.2.3 ENUM_MAP_PAIR

```
typedef ENUM_MAP::value_type Rcpp::class_Base::ENUM_MAP_PAIR
```

Definition at line 90 of file class_Base.h.

6.69.2.4 XP_Class

```
typedef Rcpp::XPtr<class_Base> Rcpp::class_Base::XP_Class
```

Definition at line 28 of file class_Base.h.

6.69.3 Constructor & Destructor Documentation

6.69.3.1 class_Base() [1/2]

```
Rcpp::class_Base::class_Base ( )  [inline]
```

Definition at line 30 of file class_Base.h.

6.69.3.2 class_Base() [2/2]

```
Rcpp::class_Base::class_Base (
             const char * name_,
             const char * doc )  [inline]
```

Definition at line 31 of file class_Base.h.

6.69.3.3 ∼class_Base()

```
virtual Rcpp::class_Base::∼class_Base ( )  [inline], [virtual]
```

Definition at line 69 of file class_Base.h.

6.69.4 Member Function Documentation

6.69.4.1 add_enum()

```
void Rcpp::class_Base::add_enum (
             const std::string & enum_name,
             const std::map< std::string, int > & value )  [inline]
```

Definition at line 81 of file class_Base.h.

References enums.

Referenced by Rcpp::Module::add_enum().

6.69.4.2 complete()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::complete ( )  [inline], [virtual]
```

Definition at line 68 of file class_Base.h.

6.69.4.3 fields()

```
virtual Rcpp::List Rcpp::class_Base::fields (
             const XP_Class &  )  [inline], [virtual]
```

Definition at line 34 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.69.4.4 get_typeinfo_name()

```
virtual std::string Rcpp::class_Base::get_typeinfo_name ( )  [inline], [virtual]
```

Definition at line 77 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass(), and has_typeinfo_name().

6.69.4.5 getConstructors()

```
virtual Rcpp::List Rcpp::class_Base::getConstructors (
             const XP_Class & ,
             std::string &  )  [inline], [virtual]
```

Definition at line 36 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.69.4.6 getMethods()

```
virtual Rcpp::List Rcpp::class_Base::getMethods (
             const XP_Class & ,
             std::string &  )  [inline], [virtual]
```

Definition at line 35 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.69.4.7 getProperty()

```
virtual SEXP Rcpp::class_Base::getProperty (
             SEXP ,
             SEXP  )  [inline], [virtual]
```

Definition at line 71 of file class_Base.h.

6.69.4.8 has_default_constructor()

```
virtual bool Rcpp::class_Base::has_default_constructor ( )  [inline], [virtual]
```

Definition at line 40 of file class_Base.h.

6.69.4.9 has_method()

```
virtual bool Rcpp::class_Base::has_method (
             const std::string &  )  [inline], [virtual]
```

Definition at line 41 of file class_Base.h.

6.69.4.10 has_property()

```
virtual bool Rcpp::class_Base::has_property (
             const std::string &  )  [inline], [virtual]
```

Definition at line 44 of file class_Base.h.

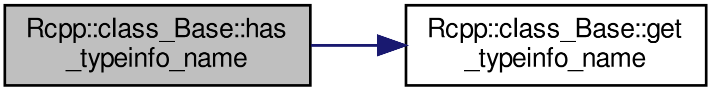
6.69.4.11 has_typeinfo_name()

```
bool Rcpp::class_Base::has_typeinfo_name (
             const std::string & name_ )  [inline]
```

Definition at line 78 of file class_Base.h.

References get_typeinfo_name().

Here is the call graph for this function:



6.69.4.12 invoke()

```
virtual SEXP Rcpp::class_Base::invoke (
             SEXP ,
             SEXP ,
             SEXP * ,
             int  )  [inline], [virtual]
```

Definition at line 50 of file class_Base.h.

6.69.4.13 invoke_notvoid()

```
virtual SEXP Rcpp::class_Base::invoke_notvoid (
             SEXP ,
             SEXP ,
             SEXP * ,
             int  )  [inline], [virtual]
```

Definition at line 56 of file class_Base.h.

6.69.4.14 invoke_void()

```
virtual SEXP Rcpp::class_Base::invoke_void (
             SEXP ,
             SEXP ,
             SEXP * ,
             int  )  [inline], [virtual]
```

Definition at line 53 of file class_Base.h.

6.69.4.15 method_names()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::method_names ( )  [inline], [virtual]
```

Definition at line 60 of file class_Base.h.

6.69.4.16 methods_arity()

```
virtual Rcpp::IntegerVector Rcpp::class_Base::methods_arity ( )  [inline], [virtual]
```

Definition at line 64 of file class_Base.h.

6.69.4.17 methods_voidness()

```
virtual Rcpp::LogicalVector Rcpp::class_Base::methods_voidness ( )  [inline], [virtual]
```

Definition at line 65 of file class_Base.h.

6.69.4.18 newInstance()

```
virtual SEXP Rcpp::class_Base::newInstance (
             SEXP * ,
             int  )  [inline], [virtual]
```

Definition at line 47 of file class_Base.h.

6.69.4.19 property_class()

```
virtual std::string Rcpp::class_Base::property_class (
             const std::string &  )  [inline], [virtual]
```

Definition at line 63 of file class_Base.h.

6.69.4.20 property_classes()

```
virtual Rcpp::List Rcpp::class_Base::property_classes ( )  [inline], [virtual]
```

Definition at line 66 of file class_Base.h.

6.69.4.21 property_is_readonly()

```
virtual bool Rcpp::class_Base::property_is_readonly (
             const std::string &  )  [inline], [virtual]
```

Definition at line 62 of file class_Base.h.

6.69.4.22 property_names()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::property_names ( )  [inline], [virtual]
```

Definition at line 61 of file class_Base.h.

6.69.4.23 run_finalizer()

```
virtual void Rcpp::class_Base::run_finalizer (
             SEXP  )  [inline], [virtual]
```

Definition at line 38 of file class_Base.h.

6.69.4.24 setProperty()

```
virtual void Rcpp::class_Base::setProperty (
             SEXP ,
             SEXP ,
             SEXP  )  [inline], [virtual]
```

Definition at line 74 of file class_Base.h.

6.69.5 Member Data Documentation

6.69.5.1 docstring

```
std::string Rcpp::class_Base::docstring
```

Definition at line 86 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.69.5.2 enums

```
ENUM_MAP Rcpp::class_Base::enums
```

Definition at line 91 of file class_Base.h.

Referenced by add_enum(), and Rcpp::CppClass::CppClass().

6.69.5.3 name

```
std::string Rcpp::class_Base::name
```

Definition at line 85 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

6.69.5.4 parents

```
std::vector<std::string> Rcpp::class_Base::parents
```

Definition at line 92 of file class_Base.h.

Referenced by Rcpp::CppClass::CppClass().

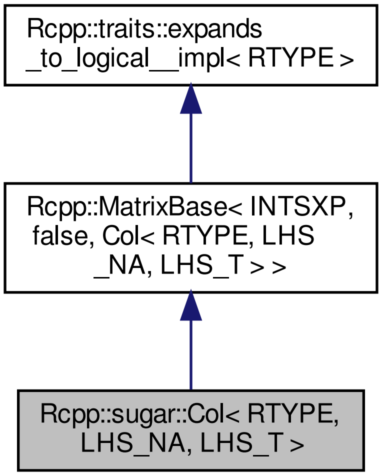
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/class_Base.h

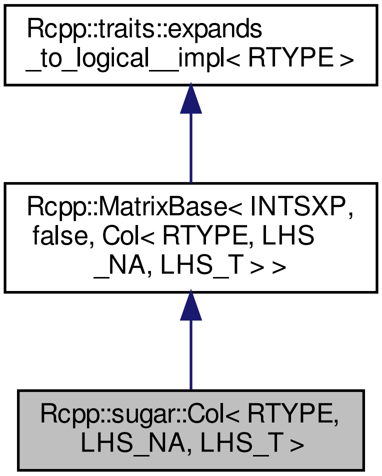
6.70 Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#include <col.h>
```

Inheritance diagram for Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >:



Collaboration diagram for Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >:



Public Types

- typedef Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE

Public Member Functions

- Col (const LHS_TYPE &lhs)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Attributes

- int nr
- int nc

Additional Inherited Members

6.70.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >

Definition at line 29 of file col.h.

6.70.2 Member Typedef Documentation

6.70.2.1 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 35 of file col.h.

6.70.3 Constructor & Destructor Documentation

6.70.3.1 Col()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::Col (
                const LHS_TYPE & lhs )  [inline]
```

Definition at line 37 of file col.h.

6.70.4 Member Function Documentation

6.70.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::ncol ( ) const  [inline]
```

Definition at line 45 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nc.

6.70.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nrow ( ) const  [inline]
```

Definition at line 44 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nr.

6.70.4.3 operator()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 39 of file col.h.

6.70.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 43 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nc, and Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >↩
::nr.

6.70.5 Member Data Documentation

6.70.5.1 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nc  [private]
```

Definition at line 48 of file col.h.

Referenced by Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::ncol(), and Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::size().

6.70.5.2 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nr  [private]
```

Definition at line 48 of file col.h.

Referenced by Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nrow(), and Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::size().

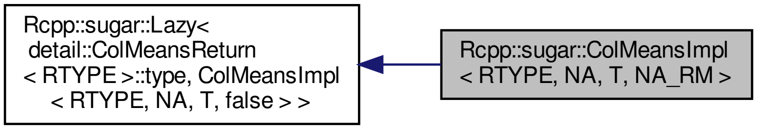
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/col.h

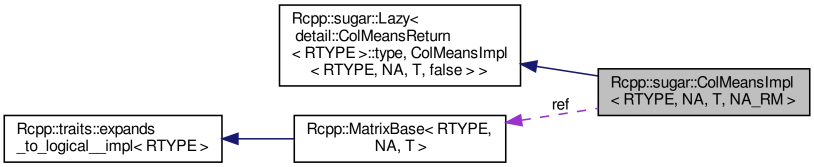
6.71 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::ColMeansReturn < RTYPE >::type, ColMeansImpl < RTYPE, NA, T, false > >
Rcpp::sugar::ColMeansImpl < RTYPE, NA, T, NA_RM >

Collaboration diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::ColMeansReturn < RTYPE >::type, ColMeansImpl < RTYPE, NA, T, false > >
Rcpp::sugar::ColMeansImpl < RTYPE, NA, T, NA_RM >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- ColMeansImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::ColMeansReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.71.1 Detailed Description

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >

Definition at line 716 of file rowSums.h.

6.71.2 Member Typedef Documentation

6.71.2.1 return_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::ColMeansReturn<RTYPE> Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::return_traits
[private]
```

Definition at line 721 of file rowSums.h.

6.71.2.2 return_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return_traits::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::return_vector  [private]
```

Definition at line 722 of file rowSums.h.

6.71.2.3 stored_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T,
NA_RM >::stored_type  [private]
```

Definition at line 723 of file rowSums.h.

6.71.3 Constructor & Destructor Documentation

6.71.3.1 ColMeansImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ColMeansImpl (
            const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 726 of file rowSums.h.

6.71.4 Member Function Documentation

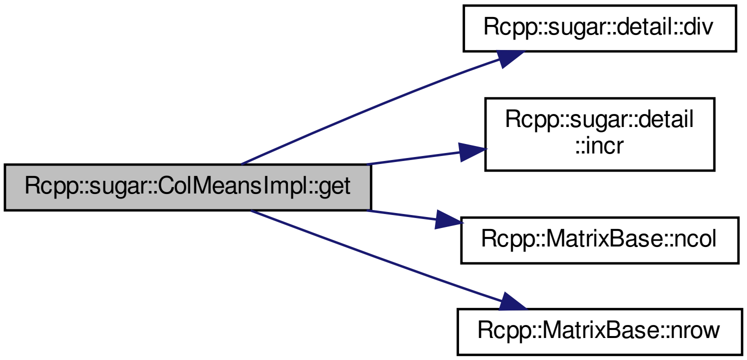
6.71.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get ( ) const  [inline]
```

Definition at line 730 of file rowSums.h.

References Rcpp::sugar::detail::div(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.71.5 Member Data Documentation

6.71.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ref  [private]
```

Definition at line 719 of file rowSums.h.

Referenced by Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

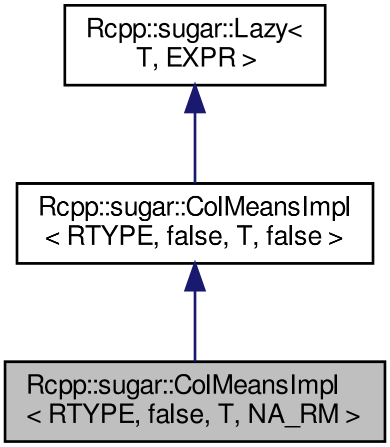
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

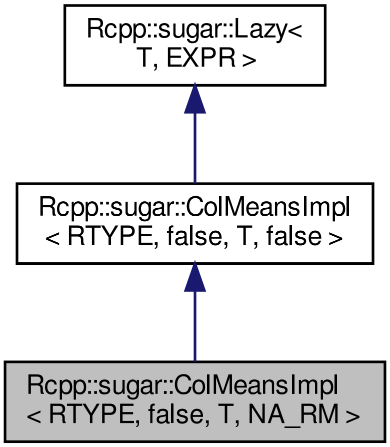
6.72 Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, NA_RM >

Collaboration diagram for Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, NA_RM >

Additional Inherited Members

6.72.1 Detailed Description

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >

Definition at line 915 of file rowSums.h.

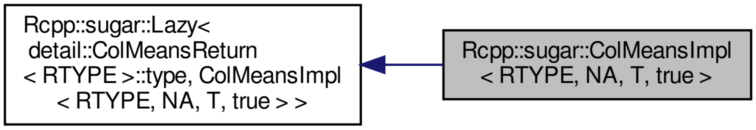
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

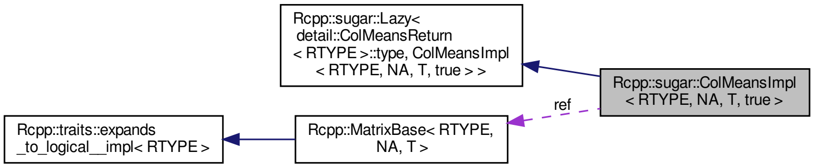
6.73 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::ColMeansReturn < RTYPE >::type, ColMeansImpl < RTYPE, NA, T, true > >
Rcpp::sugar::ColMeansImpl < RTYPE, NA, T, true >

Collaboration diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::ColMeansReturn < RTYPE >::type, ColMeansImpl < RTYPE, NA, T, true > >
Rcpp::sugar::ColMeansImpl < RTYPE, NA, T, true >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- ColMeansImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::ColMeansReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.73.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >

Definition at line 812 of file rowSums.h.

6.73.2 Member Typedef Documentation

6.73.2.1 return_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::ColMeansReturn<RTYPE> Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 817 of file rowSums.h.

6.73.2.2 return_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::return_vector  [private]
```

Definition at line 818 of file rowSums.h.

6.73.2.3 stored_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T,
true >::stored_type  [private]
```

Definition at line 819 of file rowSums.h.

6.73.3 Constructor & Destructor Documentation

6.73.3.1 ColMeansImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::ColMeansImpl (
            const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 822 of file rowSums.h.

6.73.4 Member Function Documentation

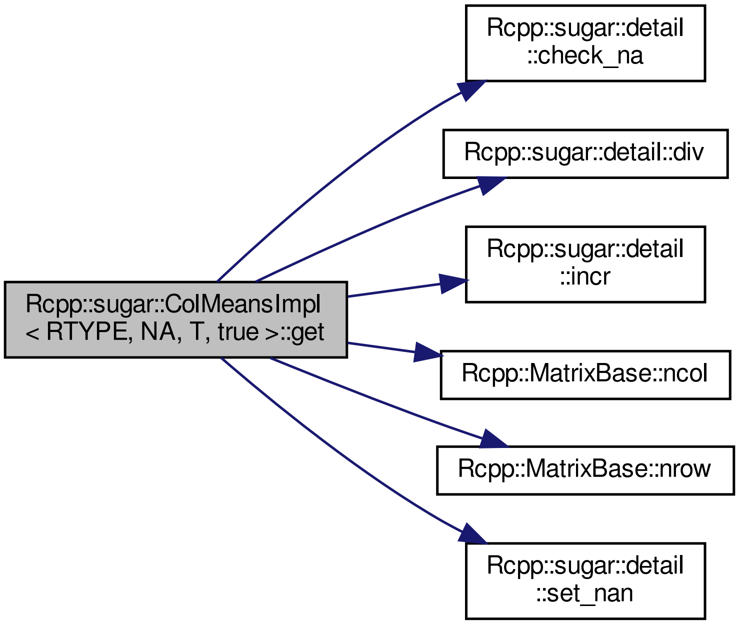
6.73.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get ( ) const  [inline]
```

Definition at line 826 of file rowSums.h.

References Rcpp::sugar::detail::check_na(), Rcpp::sugar::detail::div(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ref, and Rcpp::sugar::detail::set_nan().

Here is the call graph for this function:



6.73.5 Member Data Documentation

6.73.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::ref  [private]
```

Definition at line 815 of file rowSums.h.

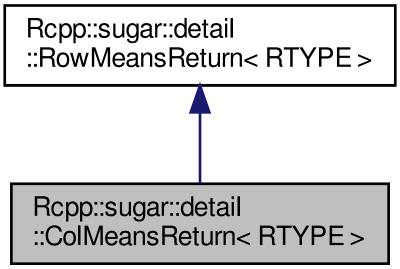
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

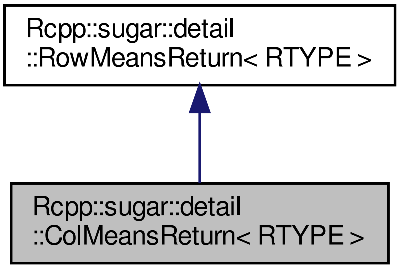
6.74 Rcpp::sugar::detail::ColMeansReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



Additional Inherited Members

6.74.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::ColMeansReturn< RTYPE >

Definition at line 115 of file rowSums.h.

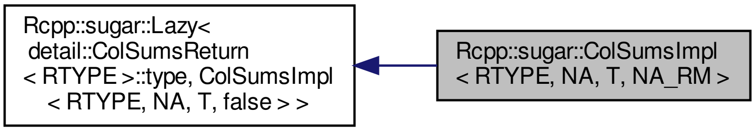
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

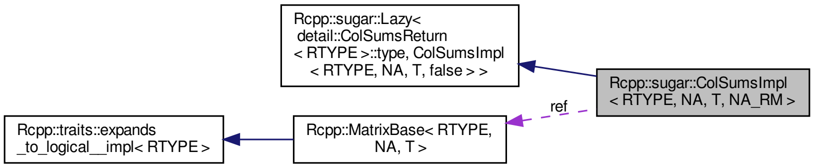
6.75 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::ColSumsReturn < RTYPE >::type, ColSumsImpl < RTYPE, NA, T, false > >
Rcpp::sugar::ColSumsImpl < RTYPE, NA, T, NA_RM >

Collaboration diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::ColSumsReturn < RTYPE >::type, ColSumsImpl < RTYPE, NA, T, false > >
Rcpp::sugar::ColSumsImpl < RTYPE, NA, T, NA_RM >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- ColSumsImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::ColSumsReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.75.1 Detailed Description

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >

Definition at line 320 of file rowSums.h.

6.75.2 Member Typedef Documentation

6.75.2.1 return_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::ColSumsReturn<RTYPE> Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::return_traits
[private]
```

Definition at line 325 of file rowSums.h.

6.75.2.2 return_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return_traits::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::return_vector  [private]
```

Definition at line 326 of file rowSums.h.

6.75.2.3 stored_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T,
NA_RM >::stored_type  [private]
```

Definition at line 327 of file rowSums.h.

6.75.3 Constructor & Destructor Documentation

6.75.3.1 ColSumsImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ColSumsImpl (
             const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 330 of file rowSums.h.

6.75.4 Member Function Documentation

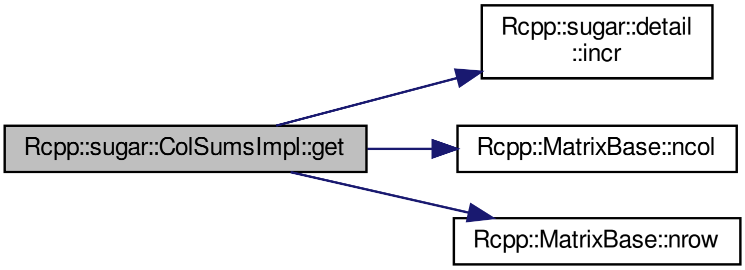
6.75.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get ( ) const  [inline]
```

Definition at line 334 of file rowSums.h.

References Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.75.5 Member Data Documentation

6.75.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref  [private]
```

Definition at line 323 of file rowSums.h.

Referenced by Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get().

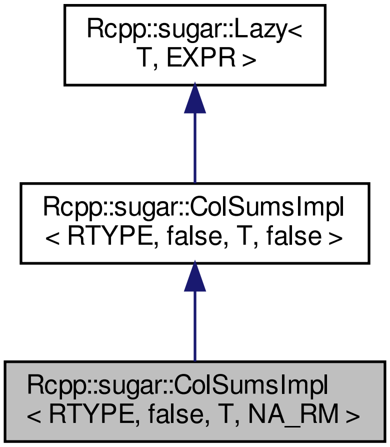
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

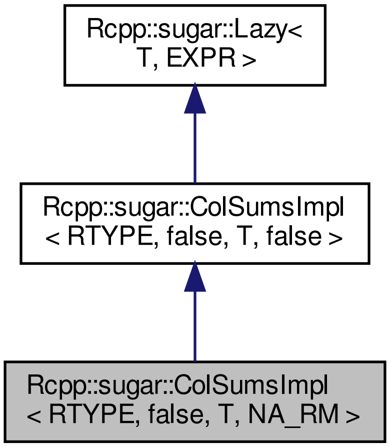
6.76 Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, NA_RM >

Collaboration diagram for Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, NA_RM >

Additional Inherited Members

6.76.1 Detailed Description

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >

Definition at line 494 of file rowSums.h.

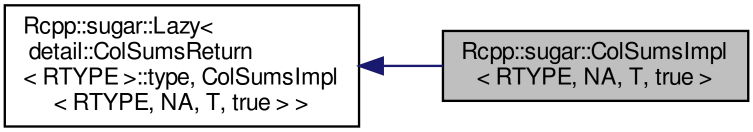
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

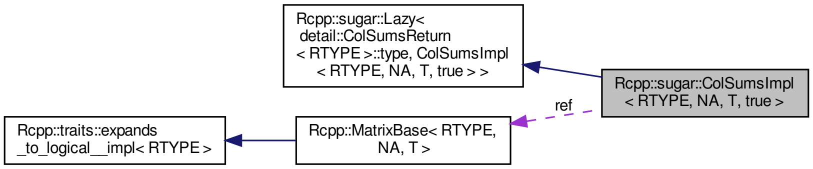
6.77 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::ColSumsReturn < RTYPE >::type, ColSumsImpl < RTYPE, NA, T, true > >
Rcpp::sugar::ColSumsImpl < RTYPE, NA, T, true >

Collaboration diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::ColSumsReturn < RTYPE >::type, ColSumsImpl < RTYPE, NA, T, true > >
Rcpp::sugar::ColSumsImpl < RTYPE, NA, T, true >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

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

6.77.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >

Definition at line 410 of file rowSums.h.

6.77.2 Member Typedef Documentation

6.77.2.1 return_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::ColSumsReturn<RTYPE> Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 415 of file rowSums.h.

6.77.2.2 return_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::return_vector  [private]
```

Definition at line 416 of file rowSums.h.

6.77.2.3 stored_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T,
true >::stored_type  [private]
```

Definition at line 417 of file rowSums.h.

6.77.3 Constructor & Destructor Documentation

6.77.3.1 ColSumsImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::ColSumsImpl (
             const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 420 of file rowSums.h.

6.77.4 Member Function Documentation

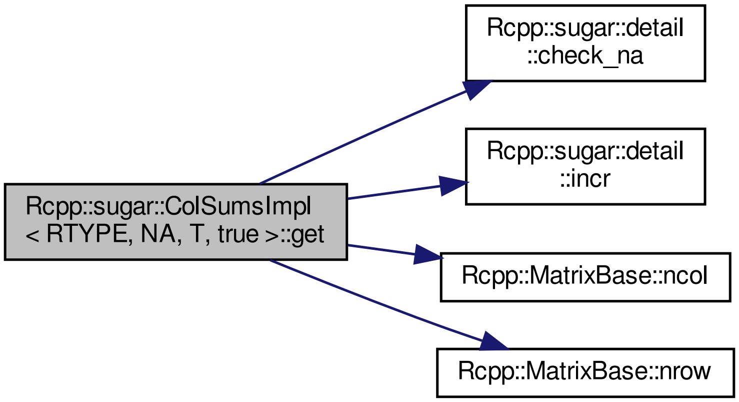
6.77.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get ( ) const  [inline]
```

Definition at line 424 of file rowSums.h.

References Rcpp::sugar::detail::check_na(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.77.5 Member Data Documentation

6.77.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::ref  [private]
```

Definition at line 413 of file rowSums.h.

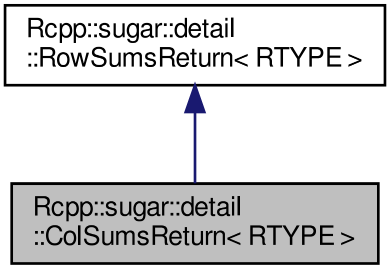
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

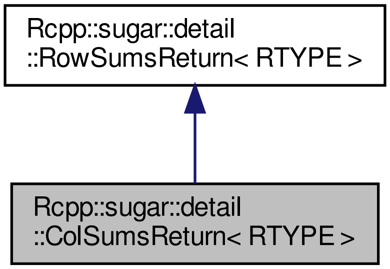
6.78 Rcpp::sugar::detail::ColSumsReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



Additional Inherited Members

6.78.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::ColSumsReturn< RTYPE >

Definition at line 98 of file rowSums.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.79 Rcpp::attributes::CommentState Class Reference

Public Member Functions

- CommentState ()
- bool inComment () const
- void submitLine (const std::string &line)
- void reset ()

Private Member Functions

- CommentState (const CommentState &)
- CommentState & operator= (const CommentState &)

Private Attributes

- bool inComment_

6.79.1 Detailed Description

Definition at line 470 of file attributes.cpp.

6.79.2 Constructor & Destructor Documentation

6.79.2.1 CommentState() [1/2]

```
Rcpp::attributes::CommentState::CommentState ( )  [inline]
```

Definition at line 472 of file attributes.cpp.

6.79.2.2 CommentState() [2/2]

```
Rcpp::attributes::CommentState::CommentState (
            const CommentState & )  [private]
```

6.79.3 Member Function Documentation

6.79.3.1 inComment()

```
bool Rcpp::attributes::CommentState::inComment ( ) const  [inline]
```

Definition at line 478 of file attributes.cpp.

References inComment_.

6.79.3.2 operator=()

```
CommentState& Rcpp::attributes::CommentState::operator= (
             const CommentState &  )  [private]
```

6.79.3.3 reset()

```
void Rcpp::attributes::CommentState::reset ( )  [inline]
```

Definition at line 480 of file attributes.cpp.

References inComment_.

6.79.3.4 submitLine()

```
void Rcpp::attributes::CommentState::submitLine (
             const std::string & line )
```

6.79.4 Member Data Documentation

6.79.4.1 inComment_

```
bool Rcpp::attributes::CommentState::inComment_ [private]
```

Definition at line 482 of file attributes.cpp.

Referenced by inComment(), and reset().

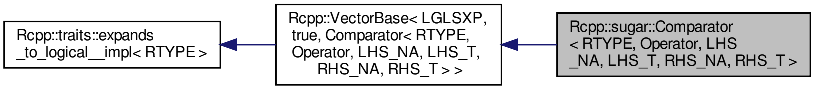
The documentation for this class was generated from the following file:

- src/attributes.cpp

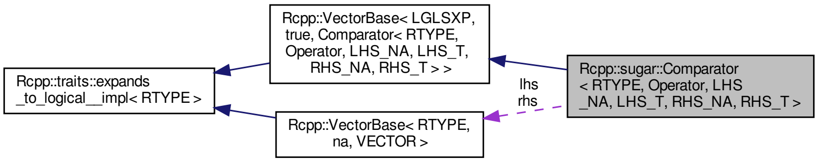
6.80 Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Comparator (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs
- Operator op

6.80.1 Detailed Description

template<int RTYPE, typename Operator, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file Comparator.h.

6.80.2 Member Typedef Documentation

6.80.2.1 LHS_TYPE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 33 of file Comparator.h.

6.80.2.2 RHS_TYPE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 34 of file Comparator.h.

6.80.2.3 STORAGE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS↩
_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 35 of file Comparator.h.

6.80.3 Constructor & Destructor Documentation

6.80.3.1 Comparator()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::Comparator (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 37 of file Comparator.h.

6.80.4 Member Function Documentation

6.80.4.1 operator[]()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
int Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 40 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::↩ Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

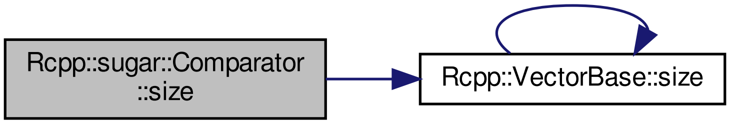
6.80.4.2 size()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::↩ VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.80.5 Member Data Documentation

6.80.5.1 lhs

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 51 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::↩ Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::size(), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::size().

6.80.5.2 op

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
Operator Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op  [private]
```

Definition at line 53 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::operator[ ]().

6.80.5.3 rhs

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 52 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::operator[ ]().

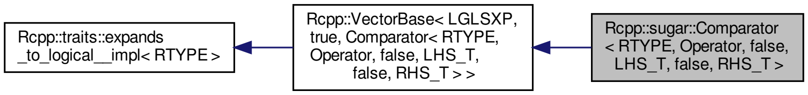
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

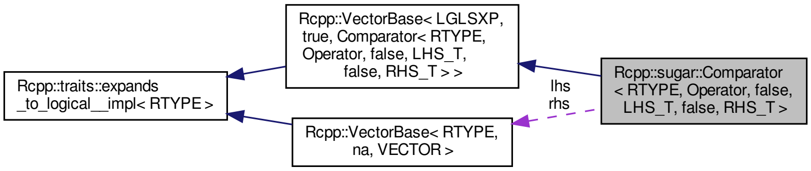
6.81 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
rhs
Rcpp::sugar::Comparator < RTYPE, Operator, false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Comparator (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs
- Operator op

6.81.1 Detailed Description

template<int RTYPE, typename Operator, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >

Definition at line 88 of file Comparator.h.

6.81.2 Member Typedef Documentation

6.81.2.1 LHS_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 92 of file Comparator.h.

6.81.2.2 RHS_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 93 of file Comparator.h.

6.81.2.3 STORAGE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 94 of file Comparator.h.

6.81.3 Constructor & Destructor Documentation

6.81.3.1 Comparator()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::Comparator (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 96 of file Comparator.h.

6.81.4 Member Function Documentation

6.81.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
int Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 99 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

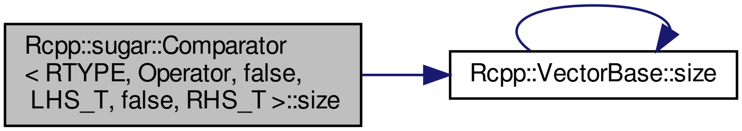
6.81.4.2 size()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 103 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.81.5 Member Data Documentation

6.81.5.1 lhs

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 106 of file Comparator.h.

6.81.5.2 op

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
Operator Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::op  [private]
```

Definition at line 108 of file Comparator.h.

6.81.5.3 rhs

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 107 of file Comparator.h.

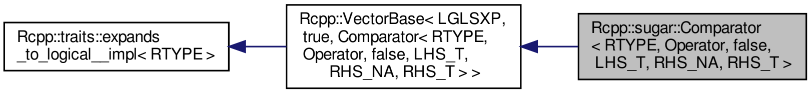
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

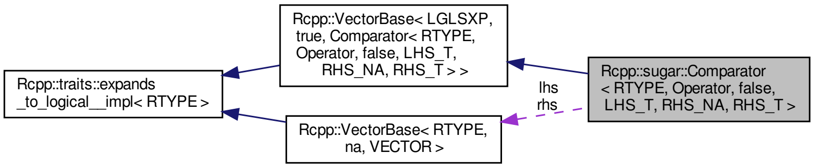
6.82 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::Comparator < RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >
lhs
rhs
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Comparator (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs
- Operator op

6.82.1 Detailed Description

template<int RTYPE, typename Operator, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >

Definition at line 60 of file Comparator.h.

6.82.2 Member Typedef Documentation

6.82.2.1 LHS_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 64 of file Comparator.h.

6.82.2.2 RHS_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 65 of file Comparator.h.

6.82.2.3 STORAGE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 66 of file Comparator.h.

6.82.3 Constructor & Destructor Documentation

6.82.3.1 Comparator()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::Comparator (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 68 of file Comparator.h.

6.82.4 Member Function Documentation

6.82.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 71 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::↩ Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

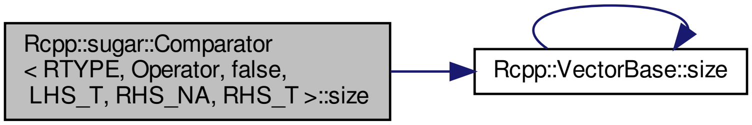
6.82.4.2 size()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 77 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::↩
VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.82.5 Member Data Documentation

6.82.5.1 lhs

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 80 of file Comparator.h.

6.82.5.2 op

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
Operator Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::op  [private]
```

Definition at line 82 of file Comparator.h.

6.82.5.3 rhs

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 81 of file Comparator.h.

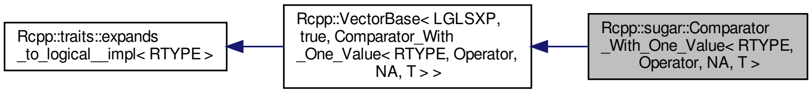
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

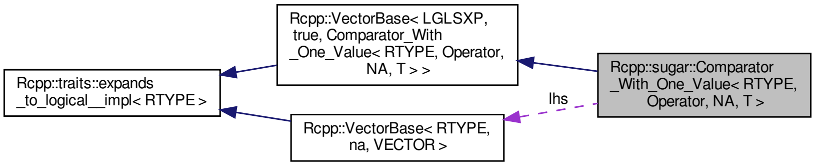
6.83 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T > Class Template Reference

```
#include <Comparator_With_One_Value.h>
```

Inheritance diagram for Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Comparator_With _One_Value< RTYPE, Operator, NA, T > >
Rcpp::sugar::Comparator _With_One_Value< RTYPE, Operator, NA, T >

Collaboration diagram for Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >:


Rcpp::VectorBase< LGLSXP, true, Comparator_With _One_Value< RTYPE, Operator, NA, T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
lhs
Rcpp::sugar::Comparator _With_One_Value< RTYPE, Operator, NA, T >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef int(Comparator_With_One_Value::∗ METHOD) (int) const

Public Member Functions

- Comparator_With_One_Value (const VEC_TYPE &lhs_, STORAGE rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Member Functions

- int rhs_is_na (int i) const
- int rhs_is_not_na (int i) const

Private Attributes

- const VEC_TYPE & lhs
- STORAGE rhs
- METHOD m
- Operator op

6.83.1 Detailed Description

template<int RTYPE, typename Operator, bool NA, typename T>
class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >

Definition at line 29 of file Comparator_With_One_Value.h.

6.83.2 Member Typedef Documentation

6.83.2.1 METHOD

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef int(Comparator_With_One_Value::* Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
NA, T >::METHOD) (int) const
```

Definition at line 33 of file Comparator_With_One_Value.h.

6.83.2.2 STORAGE

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
NA, T >::STORAGE
```

Definition at line 32 of file Comparator_With_One_Value.h.

6.83.2.3 VEC_TYPE

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA,
T >::VEC_TYPE
```

Definition at line 31 of file Comparator_With_One_Value.h.

6.83.3 Constructor & Destructor Documentation

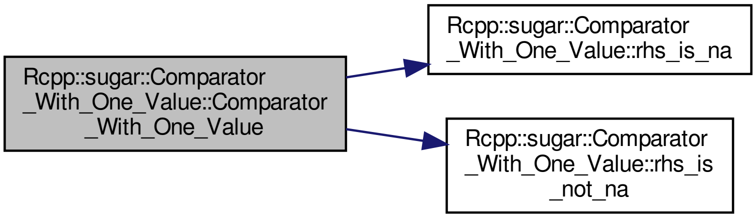
6.83.3.1 Comparator_With_One_Value()

```
template<int RTYPE, typename Operator , bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value (
            const VEC_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 35 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_na(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na().

Here is the call graph for this function:



6.83.4 Member Function Documentation

6.83.4.1 operator[]()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 44 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m.

6.83.4.2 rhs_is_na()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_na (
             int i ) const  [inline], [private]
```

Definition at line 56 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value().

6.83.4.3 rhs_is_not_na()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na (
             int i ) const  [inline], [private]
```

Definition at line 57 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::op, and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value().

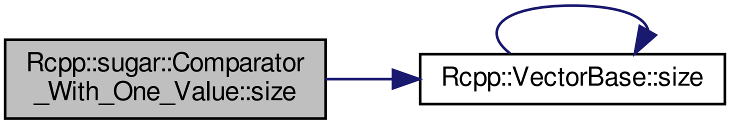
6.83.4.4 size()

```
template<int RTYPE, typename Operator , bool NA, typename T >
R_xlen_t Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::size ( ) const  [inline]
```

Definition at line 48 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.83.5 Member Data Documentation

6.83.5.1 lhs

```
template<int RTYPE, typename Operator , bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs  [private]
```

Definition at line 51 of file Comparator_With_One_Value.h.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_not_na(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::size(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size().

6.83.5.2 m

```
template<int RTYPE, typename Operator , bool NA, typename T >
METHOD Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m  [private]
```

Definition at line 53 of file Comparator_With_One_Value.h.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::operator[ ](), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::operator[ ]().

6.83.5.3 op

```
template<int RTYPE, typename Operator , bool NA, typename T >
Operator Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::op  [private]
```

Definition at line 54 of file Comparator_With_One_Value.h.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_not_na().

6.83.5.4 rhs

```
template<int RTYPE, typename Operator , bool NA, typename T >
STORAGE Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs  [private]
```

Definition at line 52 of file Comparator_With_One_Value.h.

Referenced by Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_na(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_na(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_not_na().

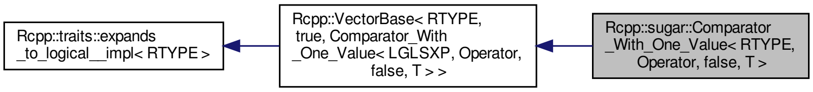
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h

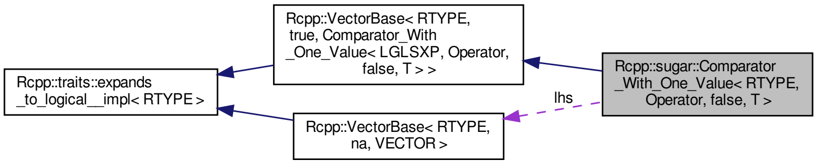
6.84 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T > Class Template Reference

```
#include <Comparator_With_One_Value.h>
```

Inheritance diagram for Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >:



Collaboration diagram for Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >:


Rcpp::VectorBase< RTYPE, true, Comparator_With _One_Value< LGLSXP, Operator, false, T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::Comparator _With_One_Value< RTYPE, Operator, false, T >
lhs_
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef int(Comparator_With_One_Value::∗ METHOD) (int) const

Public Member Functions

- Comparator_With_One_Value (const VEC_TYPE &lhs_, STORAGE rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Member Functions

- int rhs_is_na (int i) const
- int rhs_is_not_na (int i) const

Private Attributes

- const VEC_TYPE & lhs
- STORAGE rhs
- METHOD m
- Operator op

6.84.1 Detailed Description

template< int RTYPE, typename Operator, typename T>
class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >

Definition at line 66 of file Comparator_With_One_Value.h.

6.84.2 Member Typedef Documentation

6.84.2.1 METHOD

```
template<int RTYPE, typename Operator , typename T >
typedef int(Comparator_With_One_Value::∗ Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::METHOD) (int) const
```

Definition at line 72 of file Comparator_With_One_Value.h.

6.84.2.2 STORAGE

```
template<int RTYPE, typename Operator , typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::STORAGE
```

Definition at line 71 of file Comparator_With_One_Value.h.

6.84.2.3 VEC_TYPE

```
template<int RTYPE, typename Operator , typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::VEC_TYPE
```

Definition at line 70 of file Comparator_With_One_Value.h.

6.84.3 Constructor & Destructor Documentation

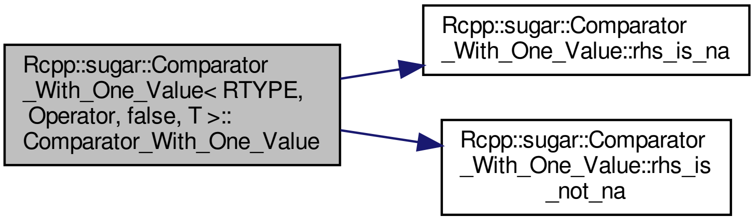
6.84.3.1 Comparator_With_One_Value()

```
template<int RTYPE, typename Operator , typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value (
            const VEC_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 74 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_na(), and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na().

Here is the call graph for this function:



6.84.4 Member Function Documentation

6.84.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 83 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m.

6.84.4.2 rhs_is_na()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_na (
             int i ) const  [inline], [private]
```

Definition at line 95 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs.

6.84.4.3 rhs_is_not_na()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_not_na (
             int i ) const  [inline], [private]
```

Definition at line 96 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs, Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::op, and Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs.

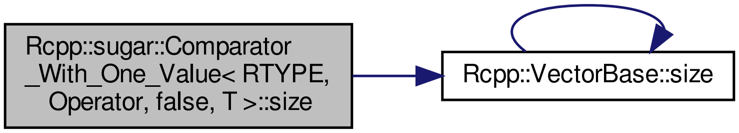
6.84.4.4 size()

```
template<int RTYPE, typename Operator , typename T >
R_xlen_t Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size ( ) const
[inline]
```

Definition at line 87 of file Comparator_With_One_Value.h.

References Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.84.5 Member Data Documentation

6.84.5.1 lhs

```
template<int RTYPE, typename Operator , typename T >
const VEC_TYPE& Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::lhs  [private]
```

Definition at line 90 of file Comparator_With_One_Value.h.

6.84.5.2 m

```
template<int RTYPE, typename Operator , typename T >
METHOD Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::m  [private]
```

Definition at line 92 of file Comparator_With_One_Value.h.

6.84.5.3 op

```
template<int RTYPE, typename Operator , typename T >
Operator Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::op  [private]
```

Definition at line 93 of file Comparator_With_One_Value.h.

6.84.5.4 rhs

```
template<int RTYPE, typename Operator , typename T >
STORAGE Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs  [private]
```

Definition at line 91 of file Comparator_With_One_Value.h.

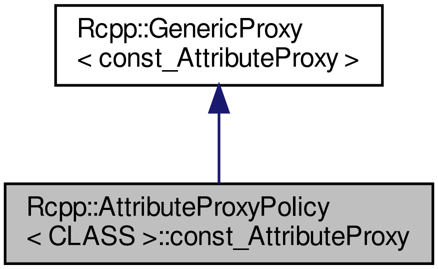
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h

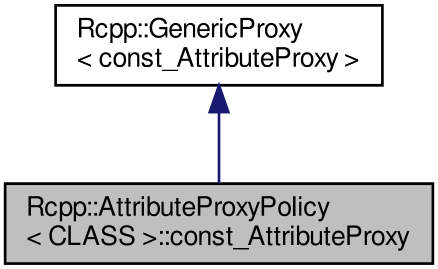
6.85 Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy Class Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy:



Collaboration diagram for Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy:



Public Member Functions

- const_AttributeProxy (const CLASS &v, const std::string &name)
- template<typename T >
  operator T () const
- operator SEXP () const

Private Member Functions

- SEXP get () const

Private Attributes

- const CLASS & parent
- SEXP attr_name

6.85.1 Detailed Description

template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy

Definition at line 56 of file AttributeProxy.h.

6.85.2 Constructor & Destructor Documentation

6.85.2.1 const_AttributeProxy()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::const_AttributeProxy (
             const CLASS & v,
             const std::string & name )  [inline]
```

Definition at line 58 of file AttributeProxy.h.

6.85.3 Member Function Documentation

6.85.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::get ( ) const  [inline], [private]
```

Definition at line 68 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::attr_name, and Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::parent.

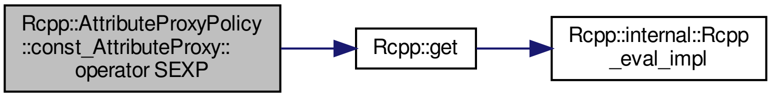
6.85.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator SEXP  [inline]
```

Definition at line 55 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



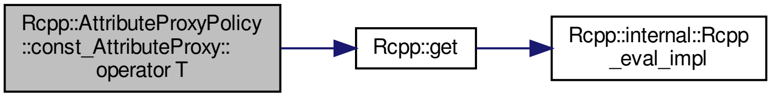
6.85.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator T
```

Definition at line 50 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.85.4 Member Data Documentation

6.85.4.1 attr_name

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::attr_name  [private]
```

Definition at line 66 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::get().

6.85.4.2 parent

```
template<typename CLASS >
const CLASS& Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::parent  [private]
```

Definition at line 65 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::get().

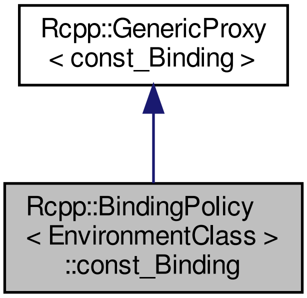
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/AttributeProxy.h
- inst/include/Rcpp/api/meat/proxy.h

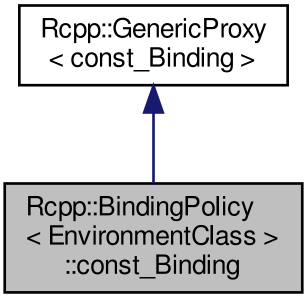
6.86 Rcpp::BindingPolicy< EnvironmentClass >::const_Binding Class Reference

```
#include <Binding.h>
```

Inheritance diagram for Rcpp::BindingPolicy< EnvironmentClass >::const_Binding:



Collaboration diagram for Rcpp::BindingPolicy< EnvironmentClass >::const_Binding:



Public Member Functions

- const_Binding (const EnvironmentClass &env_, const std::string &name_)
- bool active () const
- bool locked () const
- bool exists () const
- template<typename T >
  operator T () const

Private Member Functions

- SEXP get () const

Private Attributes

- const EnvironmentClass & env
- std::string name

6.86.1 Detailed Description

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::const_Binding

Definition at line 72 of file Binding.h.

6.86.2 Constructor & Destructor Documentation

6.86.2.1 const_Binding()

```
template<typename EnvironmentClass >
Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::const_Binding (
             const EnvironmentClass & env_,
             const std::string & name_ )  [inline]
```

Definition at line 74 of file Binding.h.

6.86.3 Member Function Documentation

6.86.3.1 active()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::active ( ) const  [inline]
```

Definition at line 77 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name.

6.86.3.2 exists()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists ( ) const  [inline]
```

Definition at line 83 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name.

6.86.3.3 get()

```
template<typename EnvironmentClass >
SEXP Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::get ( ) const  [inline], [private]
```

Definition at line 90 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name.

6.86.3.4 locked()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::locked ( ) const  [inline]
```

Definition at line 80 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name.

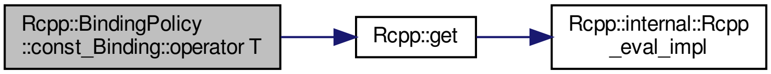
6.86.3.5 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::BindingPolicy< CLASS >::const_Binding::operator T
```

Definition at line 143 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.86.4 Member Data Documentation

6.86.4.1 env

```
template<typename EnvironmentClass >
const EnvironmentClass& Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env  [private]
```

Definition at line 94 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::locked().

6.86.4.2 name

```
template<typename EnvironmentClass >
std::string Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name  [private]
```

Definition at line 95 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::locked().

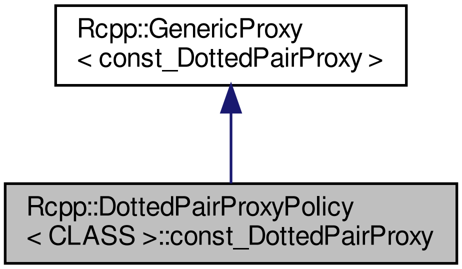
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/Binding.h
- inst/include/Rcpp/api/meat/proxy.h

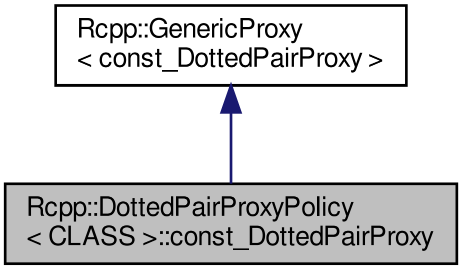
6.87 Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy Class Reference

```
#include <DottedPairProxy.h>
```

Inheritance diagram for Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy:


Rcpp::GenericProxy
< const_DottedPairProxy >
Rcpp::DottedPairProxyPolicy
< CLASS >::const_DottedPairProxy

Collaboration diagram for Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy:


Rcpp::GenericProxy
< const_DottedPairProxy >
Rcpp::DottedPairProxyPolicy
< CLASS >::const_DottedPairProxy

Public Member Functions

- const_DottedPairProxy (const CLASS &v, int index_)
- template<typename T >
  operator T () const
- SEXP get () const
- operator SEXP () const

Private Attributes

- SEXP node

6.87.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy

Definition at line 77 of file DottedPairProxy.h.

6.87.2 Constructor & Destructor Documentation

6.87.2.1 const_DottedPairProxy()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::const_DottedPairProxy (
             const CLASS & v,
             int index_ )  [inline]
```

Definition at line 79 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::node.

6.87.3 Member Function Documentation

6.87.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::get ( ) const  [inline]
```

Definition at line 93 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator SEXP().

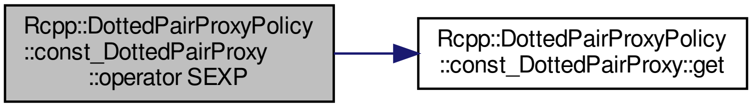
6.87.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator SEXP ( ) const  [inline]
```

Definition at line 96 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::get().

Here is the call graph for this function:


Rcpp::DottedPairProxyPolicy
::const_DottedPairProxy
::operator SEXP
Rcpp::DottedPairProxyPolicy
::const_DottedPairProxy::get

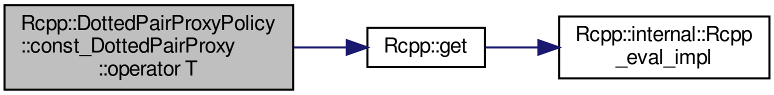
6.87.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator T
```

Definition at line 171 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.87.4 Member Data Documentation

6.87.4.1 node

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::node  [private]
```

Definition at line 101 of file DottedPairProxy.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::const_DottedPairProxy(), and Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::get().

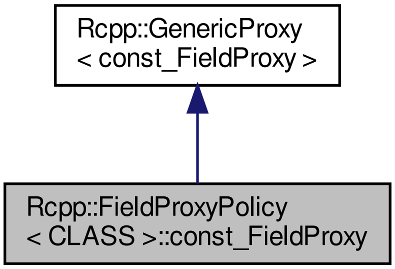
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/DottedPairProxy.h
- inst/include/Rcpp/api/meat/proxy.h

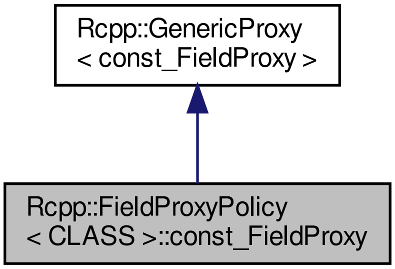
6.88 Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy Class Reference

```
#include <FieldProxy.h>
```

Inheritance diagram for Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy:



Collaboration diagram for Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy:



Public Member Functions

- const_FieldProxy (const CLASS &v, const std::string &name)
- template<typename T >
  operator T () const
- operator SEXP () const

Private Member Functions

- SEXP get () const

Private Attributes

- const CLASS & parent
- const std::string & field_name

6.88.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy

Definition at line 56 of file FieldProxy.h.

6.88.2 Constructor & Destructor Documentation

6.88.2.1 const_FieldProxy()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::const_FieldProxy (
            const CLASS & v,
            const std::string & name )  [inline]
```

Definition at line 58 of file FieldProxy.h.

6.88.3 Member Function Documentation

6.88.3.1 get()

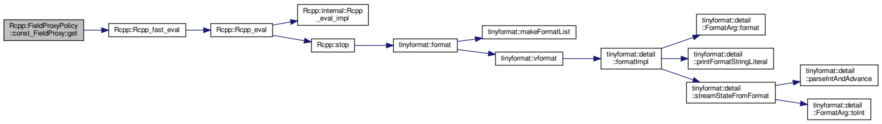
```
template<typename CLASS >
SEXP Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get ( ) const  [inline], [private]
```

Definition at line 70 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::field_name, Rcpp::FieldProxyPolicy< CLASS >↩
::const_FieldProxy::parent, and Rcpp::Rcpp_fast_eval().

Referenced by Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator SEXP().

Here is the call graph for this function:



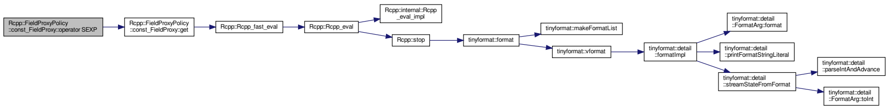
6.88.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator SEXP ( ) const  [inline]
```

Definition at line 62 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get().

Here is the call graph for this function:



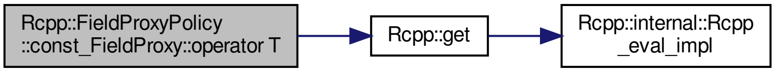
6.88.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator T
```

Definition at line 199 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.88.4 Member Data Documentation

6.88.4.1 field_name

```
template<typename CLASS >
const std::string& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::field_name  [private]
```

Definition at line 68 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get().

6.88.4.2 parent

```
template<typename CLASS >
const CLASS& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::parent  [private]
```

Definition at line 67 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::get().

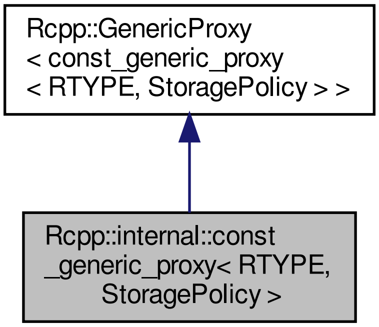
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/FieldProxy.h
- inst/include/Rcpp/api/meat/proxy.h

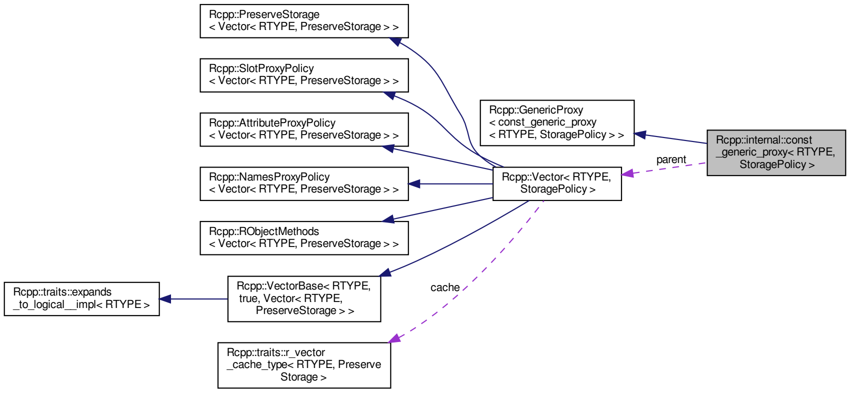
6.89 Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <const_generic_proxy.h>
```

Inheritance diagram for Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >:


Rcpp::GenericProxy < const_generic_proxy < RTYPE, StoragePolicy > >
Rcpp::internal::const _generic_proxy< RTYPE, StoragePolicy >

Collaboration diagram for Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::GenericProxy < const_generic_proxy < RTYPE, StoragePolicy > >
Rcpp::internal::const _generic_proxy< RTYPE, StoragePolicy >
parent
Rcpp::Vector< RTYPE, StoragePolicy >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
cache
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR

Public Member Functions

- const_generic_proxy ()
- const_generic_proxy (const const_generic_proxy &other)
- const_generic_proxy (const VECTOR &v, R_xlen_t i)
- operator SEXP () const
- template<typename U >
  operator U () const
- operator bool () const
- operator int () const
- void move (R_xlen_t n)

Public Attributes

- const VECTOR * parent
- R_xlen_t index

Private Member Functions

- SEXP get () const

6.89.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >

Definition at line 27 of file const_generic_proxy.h.

6.89.2 Member Typedef Documentation

6.89.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::const_generic_proxy< RTYPE, Storage↩
Policy >::VECTOR
```

Definition at line 29 of file const_generic_proxy.h.

6.89.3 Constructor & Destructor Documentation

6.89.3.1 const_generic_proxy() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy ( )  [inline]
```

Definition at line 31 of file const_generic_proxy.h.

6.89.3.2 const_generic_proxy() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy (
             const const_generic_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 33 of file const_generic_proxy.h.

6.89.3.3 const_generic_proxy() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy (
             const VECTOR & v,
             R_xlen_t i )  [inline]
```

Definition at line 36 of file const_generic_proxy.h.

6.89.4 Member Function Documentation

6.89.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline], [private]
```

Definition at line 57 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator bool(), Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator int(), Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator SEXP(), and Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator U().

6.89.4.2 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::move (
            R_xlen_t n ) [inline]
```

Definition at line 50 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::index.

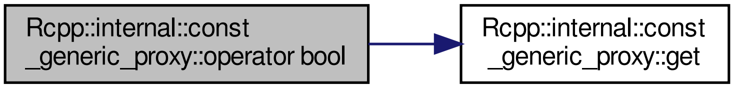
6.89.4.3 operator bool()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator bool ( ) const [inline]
```

Definition at line 47 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



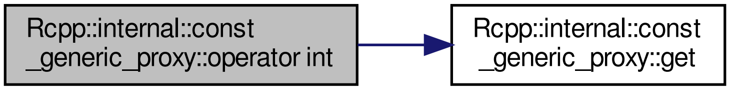
6.89.4.4 operator int()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator int ( ) const [inline]
```

Definition at line 48 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



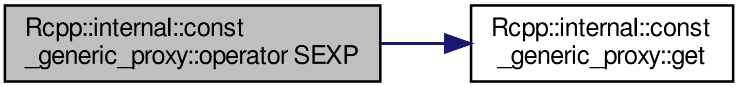
6.89.4.5 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

Definition at line 38 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



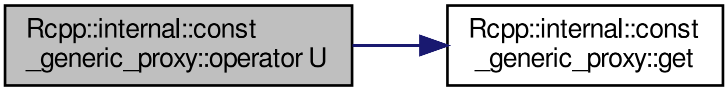
6.89.4.6 operator U()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename U >
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator U ( ) const  [inline]
```

Definition at line 42 of file const_generic_proxy.h.

References Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



6.89.5 Member Data Documentation

6.89.5.1 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 53 of file const_generic_proxy.h.

Referenced by Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::move().

6.89.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
const VECTOR* Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 52 of file const_generic_proxy.h.

Referenced by Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/const_generic_proxy.h

6.90 Rcpp::MatrixRow< RTYPE >::const_iter_traits Struct Reference

```
#include <MatrixRow.h>
```

Public Types

- typedef traits::r_vector_const_iterator< RTYPE >::type vector_iterator
- typedef int difference_type
- typedef traits::r_vector_proxy< RTYPE >::type value_type
- typedef value_type reference
- typedef std::iterator_traits< vector_iterator >::pointer pointer

6.90.1 Detailed Description

template<int RTYPE>
struct Rcpp::MatrixRow< RTYPE >::const_iter_traits

Definition at line 51 of file MatrixRow.h.

6.90.2 Member Typedef Documentation

6.90.2.1 difference_type

```
template<int RTYPE>
typedef int Rcpp::MatrixRow< RTYPE >::const_iter_traits::difference_type
```

Definition at line 55 of file MatrixRow.h.

6.90.2.2 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector_iterator>::pointer Rcpp::MatrixRow< RTYPE >::const_iter_traits::pointer
```

Definition at line 58 of file MatrixRow.h.

6.90.2.3 reference

```
template<int RTYPE>
typedef value_type Rcpp::MatrixRow< RTYPE >::const_iter_traits::reference
```

Definition at line 57 of file MatrixRow.h.

6.90.2.4 value_type

```
template<int RTYPE>
typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::const_iter_traits::value_type
```

Definition at line 56 of file MatrixRow.h.

6.90.2.5 vector_iterator

```
template<int RTYPE>
typedef traits::r_vector_const_iterator<RTYPE>::type Rcpp::MatrixRow< RTYPE >::const_iter_traits::vector_iterato
```

Definition at line 53 of file MatrixRow.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.91 Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits Struct Reference

```
#include <VectorBase.h>
```

Public Types

- typedef stored_type reference
- typedef stored_type const ∗ pointer
- typedef R_xlen_t difference_type
- typedef const stored_type value_type
- typedef std::random_access_iterator_tag iterator_category

6.91.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits

Definition at line 65 of file VectorBase.h.

6.91.2 Member Typedef Documentation

6.91.2.1 difference_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::difference_type
```

Definition at line 69 of file VectorBase.h.

6.91.2.2 iterator_category

```
template<int RTYPE, bool na, typename VECTOR >
typedef std::random_access_iterator_tag Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::iterator_catego
```

Definition at line 71 of file VectorBase.h.

6.91.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type const* Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::pointer
```

Definition at line 68 of file VectorBase.h.

6.91.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::reference
```

Definition at line 67 of file VectorBase.h.

6.91.2.5 value_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef const stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::value_type
```

Definition at line 70 of file VectorBase.h.

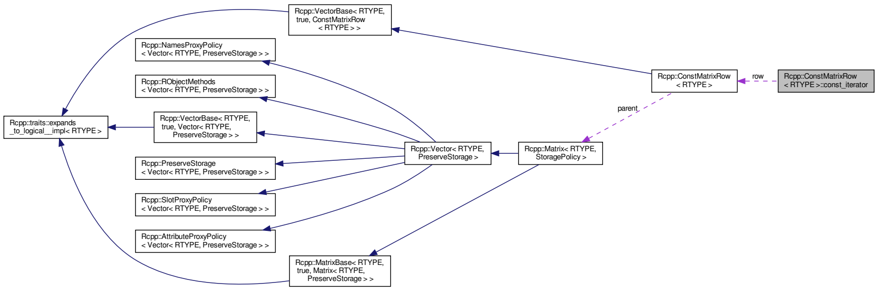
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

6.92 Rcpp::ConstMatrixRow< RTYPE >::const_iterator Class Reference

`#include <MatrixRow.h>`

Collaboration diagram for Rcpp::ConstMatrixRow< RTYPE >::const_iterator:


Rcpp::VectorBase< RTYPE, true, ConstMatrixRow < RTYPE > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
parent
Rcpp::ConstMatrixRow < RTYPE >
row
Rcpp::ConstMatrixRow < RTYPE >::const_iterator

Public Types

- typedef traits::r_vector_iterator< RTYPE >::type vector_iterator
- typedef int difference_type
- typedef traits::r_vector_const_proxy< RTYPE >::type value_type
- typedef traits::r_vector_const_proxy< RTYPE >::type reference
- typedef std::iterator_traits< vector_iterator >::pointer pointer
- typedef std::random_access_iterator_tag iterator_category

Public Member Functions

- const_iterator (const const_iterator &other)
- const_iterator (const ConstMatrixRow &row_, int index_)
- const_iterator & operator++ ()
- const_iterator operator++ (int)
- const_iterator & operator-- ()
- const_iterator operator-- (int)
- const_iterator operator+ (difference_type n) const
- const_iterator operator- (difference_type n) const
- difference_type operator- (const const_iterator &other) const
- const_iterator & operator+= (difference_type n)
- const_iterator & operator-= (difference_type n)
- const reference operator∗ ()
- const pointer operator-> ()
- bool operator== (const const_iterator &other)
- bool operator!= (const const_iterator &other)
- bool operator< (const const_iterator &other)
- bool operator> (const const_iterator &other)
- bool operator<= (const const_iterator &other)
- bool operator>= (const const_iterator &other)
- const reference operator[ ] (int i) const
- difference_type operator- (const const_iterator &other)

Private Attributes

- const ConstMatrixRow & row
- int index

6.92.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >::const_iterator

Definition at line 222 of file MatrixRow.h.

6.92.2 Member Typedef Documentation

6.92.2.1 difference_type

```
template<int RTYPE>
typedef int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::difference_type
```

Definition at line 226 of file MatrixRow.h.

6.92.2.2 iterator_category

```
template<int RTYPE>
typedef std::random_access_iterator_tag Rcpp::ConstMatrixRow< RTYPE >::const_iterator::iterator↩
_category
```

Definition at line 231 of file MatrixRow.h.

6.92.2.3 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector_iterator>::pointer Rcpp::ConstMatrixRow< RTYPE >::const_↩
iterator::pointer
```

Definition at line 229 of file MatrixRow.h.

6.92.2.4 reference

```
template<int RTYPE>
typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↩
::reference
```

Definition at line 228 of file MatrixRow.h.

6.92.2.5 value_type

```
template<int RTYPE>
typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↩
::value_type
```

Definition at line 227 of file MatrixRow.h.

6.92.2.6 vector_iterator

```
template<int RTYPE>
typedef traits::r_vector_iterator<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↩
::vector_iterator
```

Definition at line 224 of file MatrixRow.h.

6.92.3 Constructor & Destructor Documentation

6.92.3.1 const_iterator() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (
             const const_iterator & other )  [inline]
```

Definition at line 233 of file MatrixRow.h.

6.92.3.2 const_iterator() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (
             const ConstMatrixRow & row_,
             int index_ )  [inline]
```

Definition at line 234 of file MatrixRow.h.

6.92.4 Member Function Documentation

6.92.4.1 operator"!=()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator!= (
             const const_iterator & other )  [inline]
```

Definition at line 271 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.2 operator*()

```
template<int RTYPE>
const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator* ( )  [inline]
```

Definition at line 263 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::row.

6.92.4.3 operator+()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+ (
             difference_type n ) const  [inline]
```

Definition at line 256 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::row.

6.92.4.4 operator++() [1/2]

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ ( )  [inline]
```

Definition at line 236 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.5 operator++() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ (
             int  )  [inline]
```

Definition at line 240 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.6 operator+=()

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+= (
             difference_type n )  [inline]
```

Definition at line 260 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.7 operator-() [1/3]

```
template<int RTYPE>
difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
             const const_iterator & other )  [inline]
```

Definition at line 281 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.8 operator-() [2/3]

```
template<int RTYPE>
difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
             const const_iterator & other ) const  [inline]
```

Definition at line 258 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.9 operator-() [3/3]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
             difference_type n ) const  [inline]
```

Definition at line 257 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::row.

6.92.4.10 operator--() [1/2]

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- ( )  [inline]
```

Definition at line 246 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.11 operator--() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- (
             int  )  [inline]
```

Definition at line 250 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.12 operator-=()

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-= (
            difference_type n )  [inline]
```

Definition at line 261 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.13 operator->()

```
template<int RTYPE>
const pointer Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-> ( )  [inline]
```

Definition at line 266 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::row.

6.92.4.14 operator<()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator< (
            const const_iterator & other )  [inline]
```

Definition at line 272 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.15 operator<=()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<= (
            const const_iterator & other )  [inline]
```

Definition at line 274 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.16 operator==()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator== (
             const const_iterator & other )  [inline]
```

Definition at line 270 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.17 operator>()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator> (
             const const_iterator & other )  [inline]
```

Definition at line 273 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.18 operator>=()

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>= (
             const const_iterator & other )  [inline]
```

Definition at line 275 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index.

6.92.4.19 operator[]()

```
template<int RTYPE>
const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[] (
             int i ) const  [inline]
```

Definition at line 277 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::row.

6.92.5 Member Data Documentation

6.92.5.1 index

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index [private]
```

Definition at line 287 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator!=(), Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::operator∗(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+=(), Rcpp::↩ ConstMatrixRow< RTYPE >::const_iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator--(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-=(), Rcpp::ConstMatrixRow< RTYPE >::const↩ _iterator::operator->(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<=(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator==(), Rcpp↩ ::ConstMatrixRow< RTYPE >::const_iterator::operator>(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator↩ ::operator>=(), and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[ ]().

6.92.5.2 row

```
template<int RTYPE>
const ConstMatrixRow& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::row [private]
```

Definition at line 286 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator∗(), Rcpp::ConstMatrixRow< RTYPE >↩ ::const_iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator->(), and Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[ ]().

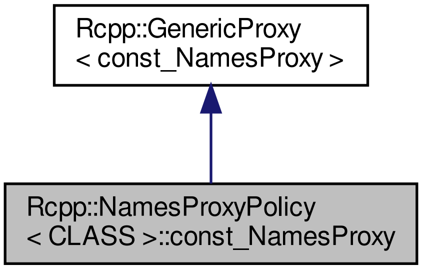
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

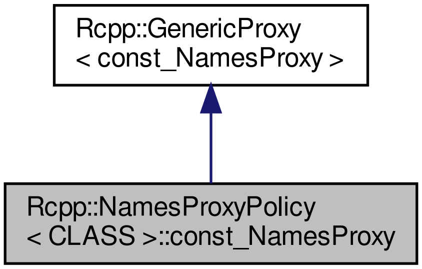
6.93 Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy Class Reference

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy:



Collaboration diagram for Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy:



Public Member Functions

- const_NamesProxy (const CLASS &v)
- template<typename T >
  operator T () const

Private Member Functions

- SEXP get () const

Private Attributes

- const CLASS & parent

6.93.1 Detailed Description

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy

Definition at line 67 of file NamesProxy.h.

6.93.2 Constructor & Destructor Documentation

6.93.2.1 const_NamesProxy()

```
template<typename CLASS >
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::const_NamesProxy (
             const CLASS & v )  [inline]
```

Definition at line 69 of file NamesProxy.h.

6.93.3 Member Function Documentation

6.93.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::get ( ) const  [inline], [private]
```

Definition at line 76 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::parent, and RCPP_GET_NAMES.

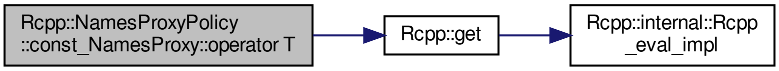
6.93.3.2 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::operator T
```

Definition at line 76 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:


Rcpp::NamesProxyPolicy ::const_NamesProxy::operator T
Rcpp::get
Rcpp::internal::Rcpp _eval_impl

6.93.4 Member Data Documentation

6.93.4.1 parent

```
template<typename CLASS >
const CLASS& Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::parent  [private]
```

Definition at line 74 of file NamesProxy.h.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::get().

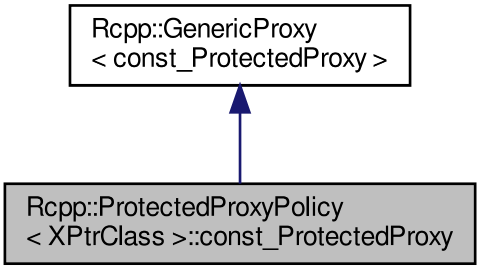
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/NamesProxy.h
- inst/include/Rcpp/api/meat/proxy.h

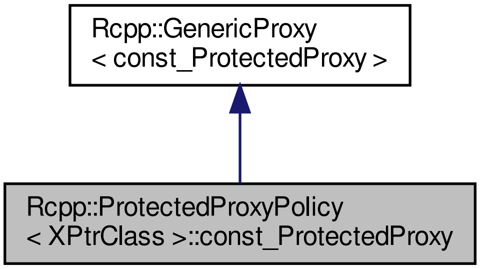
6.94 Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy Class Reference

```
#include <ProtectedProxy.h>
```

Inheritance diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy:


Rcpp::GenericProxy
< const_ProtectedProxy >
Rcpp::ProtectedProxyPolicy
< XPtrClass >::const_ProtectedProxy

Collaboration diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy:


Rcpp::GenericProxy
< const_ProtectedProxy >
Rcpp::ProtectedProxyPolicy
< XPtrClass >::const_ProtectedProxy

Public Member Functions

- const_ProtectedProxy (const XPtrClass &xp_)
- template<typename U >
  operator U () const
- operator SEXP () const

Private Member Functions

- SEXP get () const

Private Attributes

- const XPtrClass & xp

6.94.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy

Definition at line 61 of file ProtectedProxy.h.

6.94.2 Constructor & Destructor Documentation

6.94.2.1 const_ProtectedProxy()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::const_ProtectedProxy (
             const XPtrClass & xp_ )  [inline]
```

Definition at line 63 of file ProtectedProxy.h.

6.94.3 Member Function Documentation

6.94.3.1 get()

```
template<class XPtrClass >
SEXP Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get ( ) const  [inline],
[private]
```

Definition at line 77 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::xp.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator SEXP(), and Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator U().

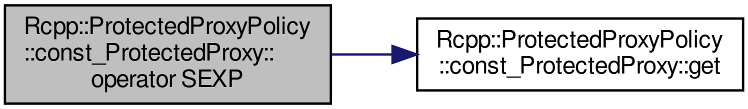
6.94.3.2 operator SEXP()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator SEXP ( ) const  [inline]
```

Definition at line 70 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get().

Here is the call graph for this function:


Rcpp::ProtectedProxyPolicy ::const_ProtectedProxy:: operator SEXP
Rcpp::ProtectedProxyPolicy ::const_ProtectedProxy::get

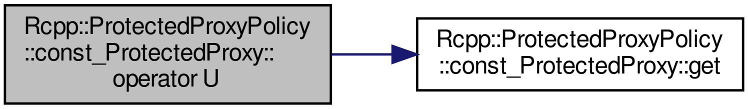
6.94.3.3 operator U()

```
template<class XPtrClass >
template<typename U >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator U ( ) const  [inline]
```

Definition at line 66 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get().

Here is the call graph for this function:


Rcpp::ProtectedProxyPolicy ::const_ProtectedProxy:: operator U
Rcpp::ProtectedProxyPolicy ::const_ProtectedProxy::get

6.94.4 Member Data Documentation

6.94.4.1 xp

```
template<class XPtrClass >
const XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::xp  [private]
```

Definition at line 75 of file ProtectedProxy.h.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get().

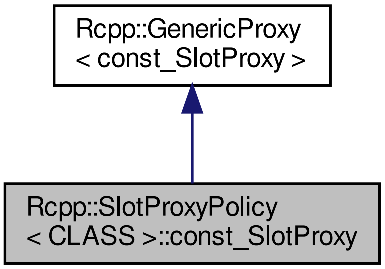
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

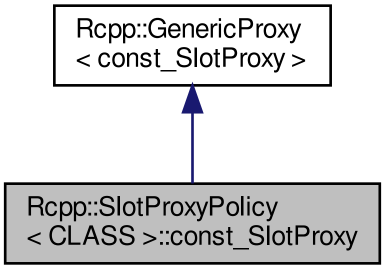
6.95 Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy Class Reference

```
#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

6.95.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy

Definition at line 59 of file SlotProxy.h.

6.95.2 Constructor & Destructor Documentation

6.95.2.1 const_SlotProxy()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::const_SlotProxy (
             const CLASS & v,
             const std::string & name )  [inline]
```

Definition at line 61 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::slot_name.

6.95.3 Member Function Documentation

6.95.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get ( ) const  [inline], [private]
```

Definition at line 78 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >↩ ::const_SlotProxy::slot_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator SEXP(), and Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator T().

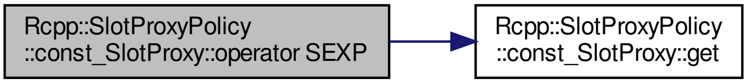
6.95.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator SEXP ( ) const  [inline]
```

Definition at line 70 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get().

Here is the call graph for this function:


Rcpp::SlotProxyPolicy ::const_SlotProxy::operator SEXP
Rcpp::SlotProxyPolicy ::const_SlotProxy::get

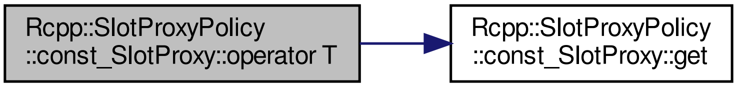
6.95.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator T ( ) const  [inline]
```

Definition at line 67 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get().

Here is the call graph for this function:


Rcpp::SlotProxyPolicy ::const_SlotProxy::operator T
Rcpp::SlotProxyPolicy ::const_SlotProxy::get

6.95.4 Member Data Documentation

6.95.4.1 parent

```
template<typename CLASS >
const CLASS& Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::parent  [private]
```

Definition at line 75 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get().

6.95.4.2 slot_name

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::slot_name  [private]
```

Definition at line 76 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::const_SlotProxy(), and Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get().

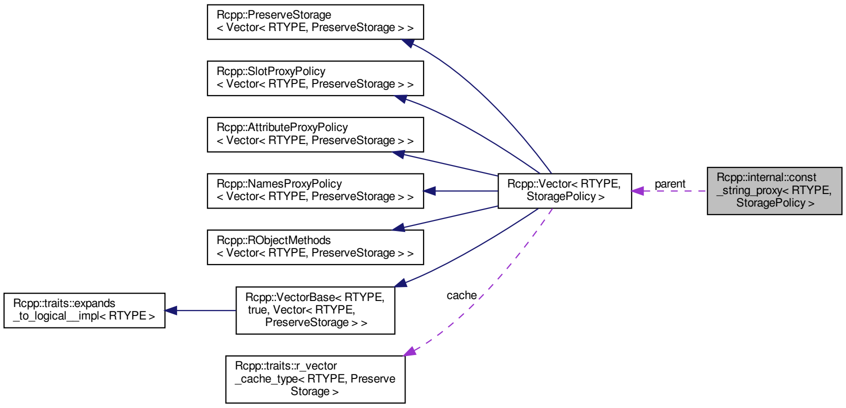
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/SlotProxy.h

6.96 Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <const_string_proxy.h>
```

Collaboration diagram for Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >:



Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef const char ∗ iterator
- typedef const char & reference

Public Member Functions

- const_string_proxy ()
- const_string_proxy (const VECTOR &v, R_xlen_t index_)
- const_string_proxy (SEXP x)
- const_string_proxy (const const_string_proxy &other)
- void import (const const_string_proxy &other)
- operator SEXP () const
- operator char ∗ () const
- void move (R_xlen_t n)
- SEXP get () const
- iterator begin () const
- iterator end () const
- R_xlen_t size () const
- bool empty () const
- reference operator[ ] (R_xlen_t n)
- bool operator== (const char ∗other)
- bool operator!= (const char ∗other)
- bool operator== (const const_string_proxy &other)
- bool operator!= (const const_string_proxy &other)
- bool operator== (SEXP other) const
- bool operator!= (SEXP other) const

Public Attributes

- const VECTOR ∗ parent
- R_xlen_t index

Static Private Attributes

- static std::string buffer

Friends

- template<int RT, template< class > class StoragePolicy_>
  std::ostream & operator<< (std::ostream &os, const const_string_proxy< RT, StoragePolicy_ > &proxy)
- template<int RT, template< class > class StoragePolicy_>
  std::string operator+ (const std::string &x, const const_string_proxy< RT, StoragePolicy_ > &proxy)

6.96.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >

Definition at line 29 of file const_string_proxy.h.

6.96.2 Member Typedef Documentation

6.96.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char* Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 33 of file const_string_proxy.h.

6.96.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char& Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 34 of file const_string_proxy.h.

6.96.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::const_string_proxy< RTYPE, Storage←
Policy >::VECTOR
```

Definition at line 32 of file const_string_proxy.h.

6.96.3 Constructor & Destructor Documentation

6.96.3.1 const_string_proxy() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy ( )  [inline]
```

Definition at line 36 of file const_string_proxy.h.

6.96.3.2 const_string_proxy() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
             const VECTOR & v,
             R_xlen_t index_ )  [inline]
```

Creates a proxy

Parameters

v	reference to the associated character vector
index	index

Definition at line 44 of file const_string_proxy.h.

6.96.3.3 const_string_proxy() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
             SEXP x )  [inline]
```

Definition at line 46 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent.

6.96.3.4 const_string_proxy() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
             const const_string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 51 of file const_string_proxy.h.

6.96.4 Member Function Documentation

6.96.4.1 begin()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin ( ) const  [inline]
```

Definition at line 94 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const_string_↩ proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::empty(), Rcpp::internal::const_string↩ _proxy< RTYPE, StoragePolicy >::end(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!=(), Rcpp::internal::operator<(), Rcpp::internal::operator<=(), Rcpp::internal::const_string_proxy< RTYPE, Storage↩ Policy >::operator==(), Rcpp::internal::operator>(), Rcpp::internal::operator>=(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator[ ](), and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::size().

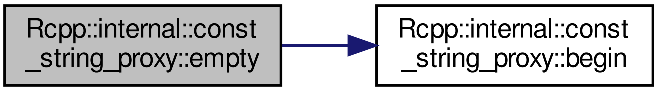
6.96.4.2 empty()

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::empty ( ) const  [inline]
```

Definition at line 97 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



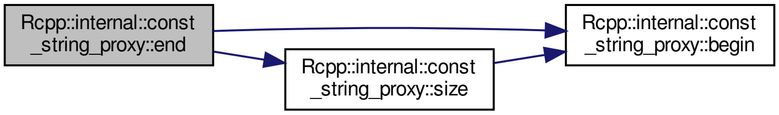
6.96.4.3 end()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::end ( ) const  [inline]
```

Definition at line 95 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.96.4.4 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline]
```

Definition at line 90 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator char *(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::String::operator!=(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::String::operator==(), and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator==().

6.96.4.5 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::import (
             const const_string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 54 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent.

6.96.4.6 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::move (
             R_xlen_t n )  [inline]
```

Definition at line 88 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index.

6.96.4.7 operator char *()

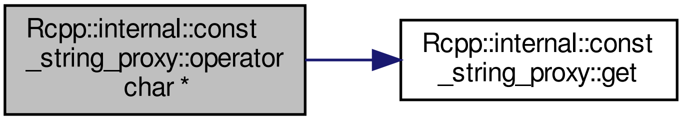
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator char * ( ) const  [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to and convert it to a C string

Definition at line 72 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:


Rcpp::internal::const _string_proxy::operator char *
Rcpp::internal::const _string_proxy::get

6.96.4.8 operator SEXP()

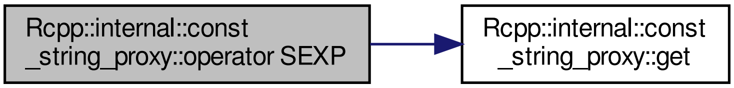
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to.

Definition at line 63 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:


Rcpp::internal::const _string_proxy::operator SEXP
Rcpp::internal::const _string_proxy::get

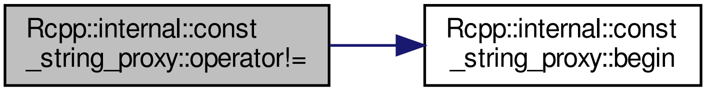
6.96.4.9 operator"!=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
             const char * other )  [inline]
```

Definition at line 103 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



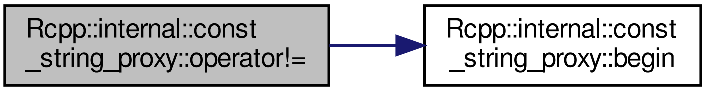
6.96.4.10 operator"!=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
             const const_string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 110 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



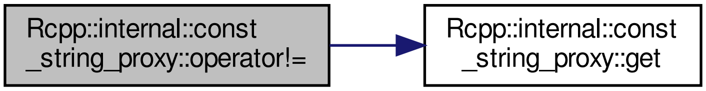
6.96.4.11 operator"!=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
            SEXP other ) const  [inline]
```

Definition at line 118 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



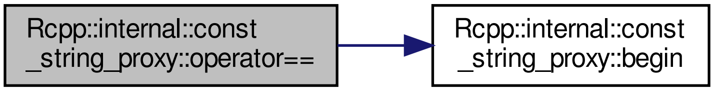
6.96.4.12 operator==() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator== (
            const char * other )  [inline]
```

Definition at line 100 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



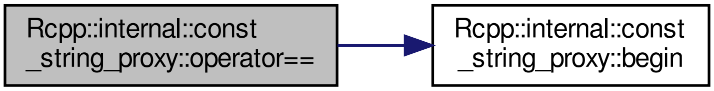
6.96.4.13 operator==() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator== (
             const const_string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 107 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



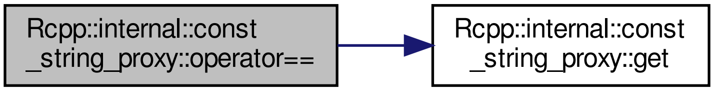
6.96.4.14 operator==() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator== (
             SEXP other ) const  [inline]
```

Definition at line 114 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



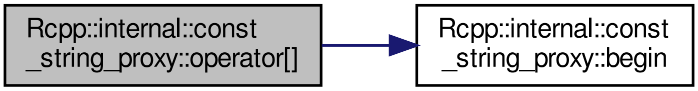
6.96.4.15 operator[]()

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator[] (
            R_xlen_t n ) [inline]
```

Definition at line 98 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.96.4.16 size()

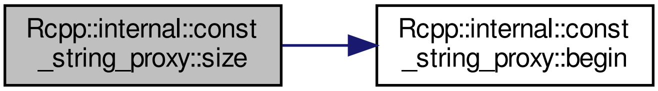
```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::size ( ) const [inline]
```

Definition at line 96 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin().

Referenced by Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



6.96.5 Friends And Related Function Documentation

6.96.5.1 operator+

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT, template< class > class StoragePolicy_>
std::string operator+ (
            const std::string & x,
            const const_string_proxy< RT, StoragePolicy_ > & proxy )  [friend]
```

6.96.5.2 operator<<

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT, template< class > class StoragePolicy_>
std::ostream& operator<< (
            std::ostream & os,
            const const_string_proxy< RT, StoragePolicy_ > & proxy )  [friend]
```

Prints the element this proxy refers to to an output stream

6.96.6 Member Data Documentation

6.96.6.1 buffer

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::buffer  [static], [private]
```

Definition at line 123 of file const_string_proxy.h.

6.96.6.2 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 87 of file const_string_proxy.h.

Referenced by Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::const_string_↩proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::import(), and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::move().

6.96.6.3 parent

```
template<int RTYPE, template< class > class StoragePolicy>
const VECTOR* Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 86 of file const_string_proxy.h.

Referenced by Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::const_string↵_proxy< RTYPE, StoragePolicy >::const_string_proxy(), Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::import().

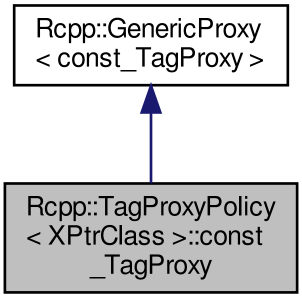
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/const_string_proxy.h

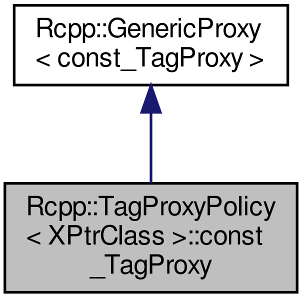
6.97 Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy Class Reference

```
#include <TagProxy.h>
```

Inheritance diagram for Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy:



Collaboration diagram for Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy:



Public Member Functions

- const_TagProxy (XPtrClass &xp_)
- template<typename U >
  operator U () const
- operator SEXP () const

Private Member Functions

- SEXP get () const

Private Attributes

- XPtrClass & xp

6.97.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy

Definition at line 53 of file TagProxy.h.

6.97.2 Constructor & Destructor Documentation

6.97.2.1 const_TagProxy()

```
template<typename XPtrClass >
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::const_TagProxy (
             XPtrClass & xp_ )  [inline]
```

Definition at line 55 of file TagProxy.h.

6.97.3 Member Function Documentation

6.97.3.1 get()

```
template<typename XPtrClass >
SEXP Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::get ( ) const  [inline], [private]
```

Definition at line 66 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::xp.

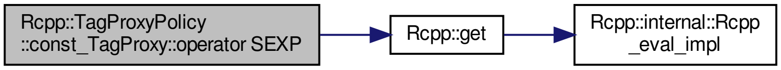
6.97.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::TagProxyPolicy< CLASS >::const_TagProxy::operator SEXP
```

Definition at line 122 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.97.3.3 operator U()

```
template<typename XPtrClass >
template<typename U >
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::operator U ( ) const
```

6.97.4 Member Data Documentation

6.97.4.1 xp

```
template<typename XPtrClass >
XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::xp  [private]
```

Definition at line 64 of file TagProxy.h.

Referenced by Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::get().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/TagProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.98 Rcpp::ConstInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef const T const_nonref

Public Member Functions

- ConstInputParameter (SEXP x_)
- operator const_nonref ()

Private Attributes

- T obj

6.98.1 Detailed Description

template<typename T>
class Rcpp::ConstInputParameter< T >

Definition at line 56 of file InputParameter.h.

6.98.2 Member Typedef Documentation

6.98.2.1 const_nonref

```
template<typename T >
typedef const T Rcpp::ConstInputParameter< T >::const_nonref
```

Definition at line 58 of file InputParameter.h.

6.98.3 Constructor & Destructor Documentation

6.98.3.1 ConstInputParameter()

```
template<typename T >
Rcpp::ConstInputParameter< T >::ConstInputParameter (
             SEXP x_ )  [inline]
```

Definition at line 59 of file InputParameter.h.

6.98.4 Member Function Documentation

6.98.4.1 operator const_nonref()

```
template<typename T >
Rcpp::ConstInputParameter< T >::operator const_nonref ( )  [inline]
```

Definition at line 61 of file InputParameter.h.

References Rcpp::ConstInputParameter< T >::obj.

6.98.5 Member Data Documentation

6.98.5.1 obj

```
template<typename T >
T Rcpp::ConstInputParameter< T >::obj  [private]
```

Definition at line 64 of file InputParameter.h.

Referenced by Rcpp::ConstInputParameter< T >::operator const_nonref().

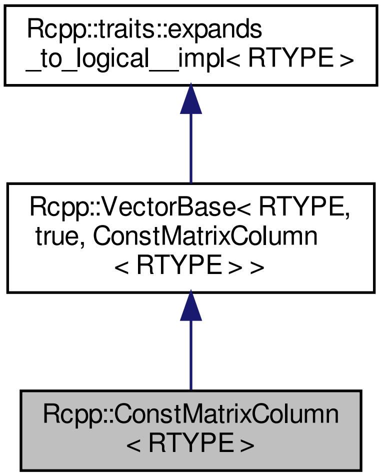
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

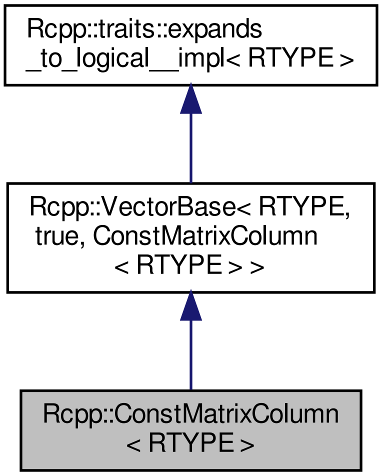
6.99 Rcpp::ConstMatrixColumn< RTYPE > Class Template Reference

```
#include <MatrixColumn.h>
```

Inheritance diagram for Rcpp::ConstMatrixColumn< RTYPE >:



Collaboration diagram for Rcpp::ConstMatrixColumn< RTYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, ConstMatrixColumn < RTYPE > >
Rcpp::ConstMatrixColumn < RTYPE >

Public Types

- typedef Matrix< RTYPE > MATRIX
- typedef MATRIX::const_Proxy const_Proxy
- typedef MATRIX::value_type value_type
- typedef MATRIX::const_iterator const_iterator

Public Member Functions

- ConstMatrixColumn (const MATRIX &parent, int i)
- ConstMatrixColumn (const ConstMatrixColumn &other)
- const_Proxy operator[ ] (int i) const
- const_iterator begin () const
- const_iterator end () const
- const_iterator cbegin () const
- const_iterator cend () const
- int size () const

Private Attributes

- const int n
- const_iterator const_start

6.99.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixColumn< RTYPE >

Definition at line 123 of file MatrixColumn.h.

6.99.2 Member Typedef Documentation

6.99.2.1 const_iterator

```
template<int RTYPE>
typedef MATRIX::const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_iterator
```

Definition at line 128 of file MatrixColumn.h.

6.99.2.2 const_Proxy

```
template<int RTYPE>
typedef MATRIX::const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::const_Proxy
```

Definition at line 126 of file MatrixColumn.h.

6.99.2.3 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::ConstMatrixColumn< RTYPE >::MATRIX
```

Definition at line 125 of file MatrixColumn.h.

6.99.2.4 value_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::ConstMatrixColumn< RTYPE >::value_type
```

Definition at line 127 of file MatrixColumn.h.

6.99.3 Constructor & Destructor Documentation

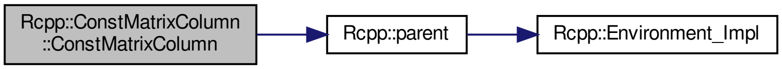
6.99.3.1 ConstMatrixColumn() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (
             const MATRIX & parent,
             int i )  [inline]
```

Definition at line 130 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:



6.99.3.2 ConstMatrixColumn() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (
             const ConstMatrixColumn< RTYPE > & other )  [inline]
```

Definition at line 141 of file MatrixColumn.h.

6.99.4 Member Function Documentation

6.99.4.1 begin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::begin ( ) const  [inline]
```

Definition at line 149 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const_start.

6.99.4.2 cbegin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::cbegin ( ) const  [inline]
```

Definition at line 157 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const_start.

6.99.4.3 cend()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::cend ( ) const  [inline]
```

Definition at line 161 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const_start, and Rcpp::ConstMatrixColumn< RTYPE >::n.

6.99.4.4 end()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::end ( ) const  [inline]
```

Definition at line 153 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const_start, and Rcpp::ConstMatrixColumn< RTYPE >::n.

6.99.4.5 operator[]()

```
template<int RTYPE>
const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::operator[] (
             int i ) const  [inline]
```

Definition at line 145 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const_start.

6.99.4.6 size()

```
template<int RTYPE>
int Rcpp::ConstMatrixColumn< RTYPE >::size ( ) const  [inline]
```

Definition at line 165 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::n.

6.99.5 Member Data Documentation

6.99.5.1 const_start

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_start  [private]
```

Definition at line 171 of file MatrixColumn.h.

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::begin(), Rcpp::ConstMatrixColumn< RTYPE >::cbegin(), Rcpp::ConstMatrixColumn< RTYPE >::cend(), Rcpp::ConstMatrixColumn< RTYPE >::end(), and Rcpp::ConstMatrixColumn< RTYPE >::operator[ ]().

6.99.5.2 n

```
template<int RTYPE>
const int Rcpp::ConstMatrixColumn< RTYPE >::n  [private]
```

Definition at line 170 of file MatrixColumn.h.

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::cend(), Rcpp::ConstMatrixColumn< RTYPE >::end(), and Rcpp::ConstMatrixColumn< RTYPE >::size().

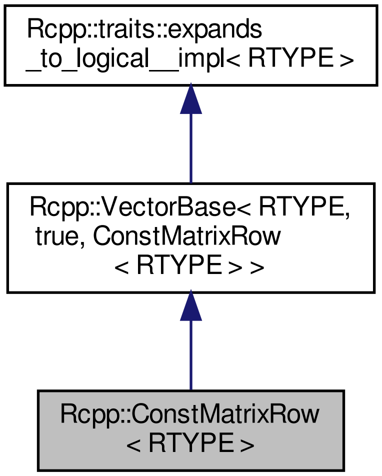
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

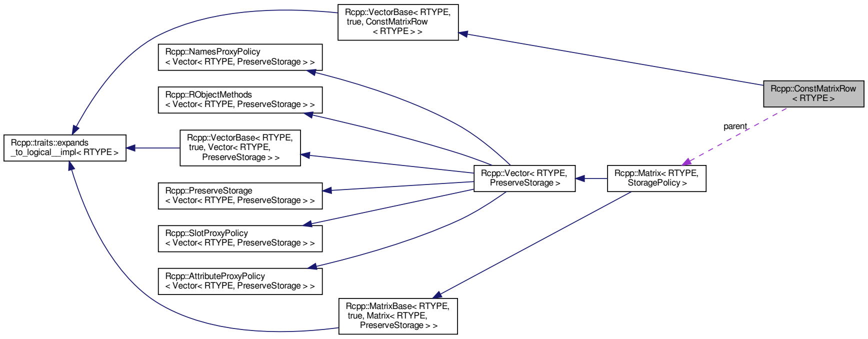
6.100 Rcpp::ConstMatrixRow< RTYPE > Class Template Reference

```
#include <MatrixRow.h>
```

Inheritance diagram for Rcpp::ConstMatrixRow< RTYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, ConstMatrixRow < RTYPE > >
Rcpp::ConstMatrixRow < RTYPE >

Collaboration diagram for Rcpp::ConstMatrixRow< RTYPE >:


Rcpp::VectorBase< RTYPE, true, ConstMatrixRow < RTYPE > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::ConstMatrixRow < RTYPE >
parent
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >

Classes

- class const_iterator

Public Types

- typedef Matrix< RTYPE > MATRIX
- typedef MATRIX::const_Proxy const_reference
- typedef MATRIX::value_type value_type
- typedef const_iterator iterator

Public Member Functions

- ConstMatrixRow (const MATRIX &object, int i)
- ConstMatrixRow (const ConstMatrixRow &other)
- const_reference operator[ ] (int i) const
- const_iterator begin () const
- const_iterator end () const
- int size () const

Private Member Functions

- int get_parent_index (int i) const

Private Attributes

- const MATRIX & parent
- MATRIX::const_iterator start
- int parent_nrow
- int row

6.100.1 Detailed Description

template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >

Definition at line 216 of file MatrixRow.h.

6.100.2 Member Typedef Documentation

6.100.2.1 const_reference

```
template<int RTYPE>
typedef MATRIX::const_Proxy Rcpp::ConstMatrixRow< RTYPE >::const_reference
```

Definition at line 219 of file MatrixRow.h.

6.100.2.2 iterator

```
template<int RTYPE>
typedef const_iterator Rcpp::ConstMatrixRow< RTYPE >::iterator
```

Definition at line 290 of file MatrixRow.h.

6.100.2.3 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::ConstMatrixRow< RTYPE >::MATRIX
```

Definition at line 218 of file MatrixRow.h.

6.100.2.4 value_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::ConstMatrixRow< RTYPE >::value_type
```

Definition at line 220 of file MatrixRow.h.

6.100.3 Constructor & Destructor Documentation

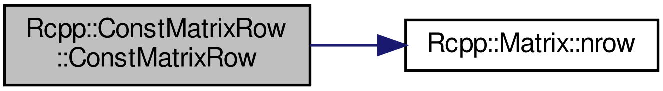
6.100.3.1 ConstMatrixRow() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (
             const MATRIX & object,
             int i ) [inline]
```

Definition at line 292 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::ConstMatrixRow< RTYPE >::parent.

Here is the call graph for this function:



6.100.3.2 ConstMatrixRow() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (
            const ConstMatrixRow< RTYPE > & other )  [inline]
```

Definition at line 305 of file MatrixRow.h.

6.100.4 Member Function Documentation

6.100.4.1 begin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::begin ( ) const  [inline]
```

Definition at line 316 of file MatrixRow.h.

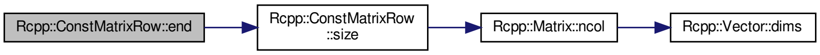
6.100.4.2 end()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::end ( ) const  [inline]
```

Definition at line 320 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::size().

Here is the call graph for this function:



6.100.4.3 get_parent_index()

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::get_parent_index (
             int i ) const  [inline], [private]
```

Definition at line 334 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::parent_nrow, and RCPP_DEBUG_4.

6.100.4.4 operator[]()

```
template<int RTYPE>
const_reference Rcpp::ConstMatrixRow< RTYPE >::operator[] (
             int i ) const  [inline]
```

Definition at line 312 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::parent, Rcpp::ConstMatrixRow< RTYPE >::parent_nrow, and Rcpp::ConstMatrixRow< RTYPE >::row.

6.100.4.5 size()

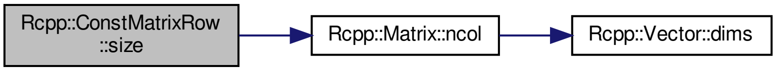
```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::size ( ) const  [inline]
```

Definition at line 324 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::ConstMatrixRow< RTYPE >::parent.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::end().

Here is the call graph for this function:



6.100.5 Member Data Documentation

6.100.5.1 parent

```
template<int RTYPE>
const MATRIX& Rcpp::ConstMatrixRow< RTYPE >::parent  [private]
```

Definition at line 329 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow(), Rcpp::ConstMatrixRow< RTYPE >::operator[ ](), and Rcpp::ConstMatrixRow< RTYPE >::size().

6.100.5.2 parent_nrow

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::parent_nrow  [private]
```

Definition at line 331 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::get_parent_index(), and Rcpp::ConstMatrixRow< RTYPE >::operator[ ]().

6.100.5.3 row

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::row  [private]
```

Definition at line 332 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::operator[ ]().

6.100.5.4 start

```
template<int RTYPE>
MATRIX::const_iterator Rcpp::ConstMatrixRow< RTYPE >::start  [private]
```

Definition at line 330 of file MatrixRow.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.101 Rcpp::ConstReferenceInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef const T & const_reference

Public Member Functions

- ConstReferenceInputParameter (SEXP x_)
- operator const_reference ()

Private Attributes

- T obj

6.101.1 Detailed Description

template<typename T>
class Rcpp::ConstReferenceInputParameter< T >

Definition at line 69 of file InputParameter.h.

6.101.2 Member Typedef Documentation

6.101.2.1 const_reference

```
template<typename T >
typedef const T& Rcpp::ConstReferenceInputParameter< T >::const_reference
```

Definition at line 71 of file InputParameter.h.

6.101.3 Constructor & Destructor Documentation

6.101.3.1 ConstReferenceInputParameter()

```
template<typename T >
Rcpp::ConstReferenceInputParameter< T >::ConstReferenceInputParameter (
             SEXP x_ )  [inline]
```

Definition at line 72 of file InputParameter.h.

6.101.4 Member Function Documentation

6.101.4.1 operator const_reference()

```
template<typename T >
Rcpp::ConstReferenceInputParameter< T >::operator const_reference ( )  [inline]
```

Definition at line 74 of file InputParameter.h.

References Rcpp::ConstReferenceInputParameter< T >::obj.

6.101.5 Member Data Documentation

6.101.5.1 obj

```
template<typename T >
T Rcpp::ConstReferenceInputParameter< T >::obj  [private]
```

Definition at line 77 of file InputParameter.h.

Referenced by Rcpp::ConstReferenceInputParameter< T >::operator const_reference().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.102 Rcpp::traits::container_exporter< Container, double > Struct Template Reference

```
#include <export.h>
```

Public Types

- typedef ContainerExporter< Container, double > type

6.102.1 Detailed Description

template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, double >

Definition at line 65 of file export.h.

6.102.2 Member Typedef Documentation

6.102.2.1 type

```
template<template< class, class > class Container>
typedef ContainerExporter< Container, double > Rcpp::traits::container_exporter< Container, double
>::type
```

Definition at line 66 of file export.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

6.103 Rcpp::traits::container_exporter< Container, int > Struct Template Reference

```
#include <export.h>
```

Public Types

- typedef ContainerExporter< Container, int > type

6.103.1 Detailed Description

template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, int >

Definition at line 62 of file export.h.

6.103.2 Member Typedef Documentation

6.103.2.1 type

```
template<template< class, class > class Container>
typedef ContainerExporter< Container, int > Rcpp::traits::container_exporter< Container, int >↩
::type
```

Definition at line 63 of file export.h.

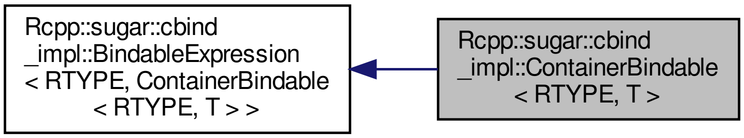
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

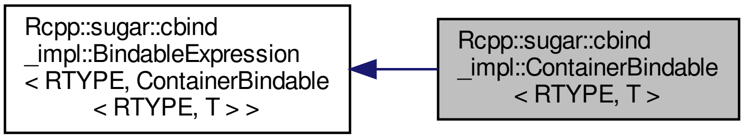
6.104 Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, ContainerBindable < RTYPE, T > >
Rcpp::sugar::cbind _impl::ContainerBindable < RTYPE, T >

Collaboration diagram for Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, ContainerBindable < RTYPE, T > >
Rcpp::sugar::cbind _impl::ContainerBindable < RTYPE, T >

Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type

Public Member Functions

- ContainerBindable (const Rcpp::Matrix< RTYPE > &vec_)
- ContainerBindable (const Rcpp::Vector< RTYPE > &vec_)
- template<typename S >
  ContainerBindable (const BindableExpression< RTYPE, S > &e)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- T vec
- R_xlen_t len
- R_xlen_t nr
- R_xlen_t nc

6.104.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >

Definition at line 82 of file cbind.h.

6.104.2 Member Typedef Documentation

6.104.2.1 stored_type

```
template<int RTYPE, typename T >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >↩
::stored_type
```

Definition at line 85 of file cbind.h.

6.104.3 Constructor & Destructor Documentation

6.104.3.1 ContainerBindable() [1/3]

```
template<int RTYPE, typename T >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
            const Rcpp::Matrix< RTYPE > & vec_ )  [inline]
```

Definition at line 92 of file cbind.h.

6.104.3.2 ContainerBindable() [2/3]

```
template<int RTYPE, typename T >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
            const Rcpp::Vector< RTYPE > & vec_ )  [inline]
```

Definition at line 97 of file cbind.h.

6.104.3.3 ContainerBindable() [3/3]

```
template<int RTYPE, typename T >
template<typename S >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
            const BindableExpression< RTYPE, S > & e )  [inline]
```

Definition at line 103 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::len, and Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec.

6.104.4 Member Function Documentation

6.104.4.1 ncol()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ncol ( ) const  [inline]
```

Definition at line 116 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nc.

6.104.4.2 nrow()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nrow ( ) const  [inline]
```

Definition at line 114 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nr.

6.104.4.3 operator()()

```
template<int RTYPE, typename T >
stored_type Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator() (
            R_xlen_t i,
            R_xlen_t j ) const  [inline]
```

Definition at line 122 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nr, and Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec.

6.104.4.4 operator[]()

```
template<int RTYPE, typename T >
stored_type Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 118 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec.

6.104.4.5 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::size ( ) const  [inline]
```

Definition at line 112 of file cbind.h.

References Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::len.

6.104.5 Member Data Documentation

6.104.5.1 len

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::len  [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable(), and Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::size().

6.104.5.2 nc

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nc  [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ncol().

6.104.5.3 nr

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nr  [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nrow(), and Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator()().

6.104.5.4 vec

```
template<int RTYPE, typename T >
T Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec  [private]
```

Definition at line 88 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable(), Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator()(), and Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.105 Rcpp::traits::ContainerExporter< ContainerTemplate, T > Class Template Reference

```
#include <export.h>
```

Public Types

- typedef ContainerTemplate< T, std::allocator< T > > Container

Public Member Functions

- ContainerExporter (SEXP x)
- ∼ContainerExporter ()
- Container get ()

Static Public Attributes

- static const int RTYPE = Rcpp::traits::r_sexptype_traits<T>::rtype

Private Attributes

- SEXP object

6.105.1 Detailed Description

template<template< class, class > class ContainerTemplate, typename T>
class Rcpp::traits::ContainerExporter< ContainerTemplate, T >

Definition at line 41 of file export.h.

6.105.2 Member Typedef Documentation

6.105.2.1 Container

```
template<template< class, class > class ContainerTemplate, typename T >
typedef ContainerTemplate<T, std::allocator<T> > Rcpp::traits::ContainerExporter< Container↩
Template, T >::Container
```

Definition at line 43 of file export.h.

6.105.3 Constructor & Destructor Documentation

6.105.3.1 ContainerExporter()

```
template<template< class, class > class ContainerTemplate, typename T >
Rcpp::traits::ContainerExporter< ContainerTemplate, T >::ContainerExporter (
            SEXP x )  [inline]
```

Definition at line 46 of file export.h.

6.105.3.2 ∼ContainerExporter()

```
template<template< class, class > class ContainerTemplate, typename T >
Rcpp::traits::ContainerExporter< ContainerTemplate, T >::∼ContainerExporter ( )  [inline]
```

Definition at line 47 of file export.h.

6.105.4 Member Function Documentation

6.105.4.1 get()

```
template<template< class, class > class ContainerTemplate, typename T >
Container Rcpp::traits::ContainerExporter< ContainerTemplate, T >::get ( )  [inline]
```

Definition at line 49 of file export.h.

References Rcpp::traits::ContainerExporter< ContainerTemplate, T >::RTYPE.

6.105.5 Member Data Documentation

6.105.5.1 object

```
template<template< class, class > class ContainerTemplate, typename T >
SEXP Rcpp::traits::ContainerExporter< ContainerTemplate, T >::object  [private]
```

Definition at line 60 of file export.h.

6.105.5.2 RTYPE

```
template<template< class, class > class ContainerTemplate, typename T >
const int Rcpp::traits::ContainerExporter< ContainerTemplate, T >::RTYPE = Rcpp::traits::r_sexptype_traits<T>↩
::rtype [static]
```

Definition at line 44 of file export.h.

Referenced by Rcpp::traits::ContainerExporter< ContainerTemplate, T >::get().

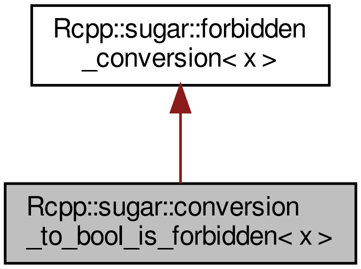
The documentation for this class was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

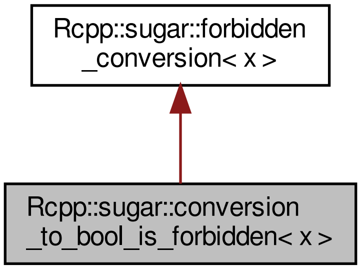
6.106 Rcpp::sugar::conversion_to_bool_is_forbidden< x > Class Template Reference

```
#include <SingleLogicalResult.h>
```

Inheritance diagram for Rcpp::sugar::conversion_to_bool_is_forbidden< x >:


Rcpp::sugar::forbidden_conversion< x >
Rcpp::sugar::conversion_to_bool_is_forbidden< x >

Collaboration diagram for Rcpp::sugar::conversion_to_bool_is_forbidden< x >:


Rcpp::sugar::forbidden_conversion< x >
Rcpp::sugar::conversion_to_bool_is_forbidden< x >

Public Member Functions

- void touch ()

6.106.1 Detailed Description

template<bool x>
class Rcpp::sugar::conversion_to_bool_is_forbidden< x >

Definition at line 35 of file SingleLogicalResult.h.

6.106.2 Member Function Documentation

6.106.2.1 touch()

```
template<bool x>
void Rcpp::sugar::conversion_to_bool_is_forbidden< x >::touch ( )  [inline]
```

Definition at line 38 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

6.107 tinyformat::detail::convertToInt< T, convertible > Struct Template Reference

```
#include <tinyformat.h>
```

Static Public Member Functions

- static int invoke (const T &)

6.107.1 Detailed Description

template<typename T, bool convertible = is_convertible<T,int>::value>
struct tinyformat::detail::convertToInt< T, convertible >

Definition at line 253 of file tinyformat.h.

6.107.2 Member Function Documentation

6.107.2.1 invoke()

```
template<typename T , bool convertible = is_convertible<T,int>::value>
static int tinyformat::detail::convertToInt< T, convertible >::invoke (
            const T & ) [inline], [static]
```

Definition at line 255 of file tinyformat.h.

References TINYFORMAT_ERROR.

Referenced by tinyformat::detail::FormatArg::toIntImpl().

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.108 tinyformat::detail::convertToInt< T, true > Struct Template Reference

```
#include <tinyformat.h>
```

Static Public Member Functions

- static int invoke (const T &value)

6.108.1 Detailed Description

template<typename T>
struct tinyformat::detail::convertToInt< T, true >

Definition at line 264 of file tinyformat.h.

6.108.2 Member Function Documentation

6.108.2.1 invoke()

```
template<typename T >
static int tinyformat::detail::convertToInt< T, true >::invoke (
             const T & value )  [inline], [static]
```

Definition at line 266 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.109 Rcpp::sugar::CountInserter< HASH, STORAGE > Class Template Reference

```
#include <table.h>
```

Public Member Functions

- CountInserter (HASH &hash_)
- void operator() (STORAGE value)

Private Attributes

- HASH & hash

6.109.1 Detailed Description

template<typename HASH, typename STORAGE>
class Rcpp::sugar::CountInserter< HASH, STORAGE >

Definition at line 29 of file table.h.

6.109.2 Constructor & Destructor Documentation

6.109.2.1 CountInserter()

```
template<typename HASH , typename STORAGE >
Rcpp::sugar::CountInserter< HASH, STORAGE >::CountInserter (
             HASH & hash_ )  [inline]
```

Definition at line 31 of file table.h.

6.109.3 Member Function Documentation

6.109.3.1 operator()()

```
template<typename HASH , typename STORAGE >
void Rcpp::sugar::CountInserter< HASH, STORAGE >::operator() (
             STORAGE value )  [inline]
```

Definition at line 33 of file table.h.

References Rcpp::sugar::CountInserter< HASH, STORAGE >::hash.

6.109.4 Member Data Documentation

6.109.4.1 hash

```
template<typename HASH , typename STORAGE >
HASH& Rcpp::sugar::CountInserter< HASH, STORAGE >::hash  [private]
```

Definition at line 38 of file table.h.

Referenced by Rcpp::sugar::CountInserter< HASH, STORAGE >::operator()().

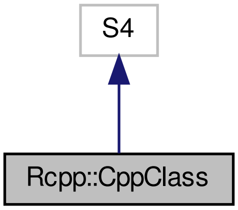
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

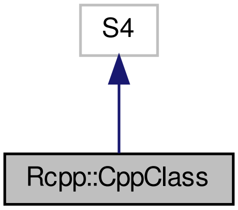
6.110 Rcpp::CppClass Class Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppClass:



Collaboration diagram for Rcpp::CppClass:



Public Types

- typedef XPtr< class_Base > XP_Class
- typedef Rcpp::XPtr< Rcpp::Module > XP

Public Member Functions

- CppClass (SEXP x)
- CppClass (Module ∗p, class_Base ∗cl, std::string &buffer)

Private Types

- typedef S4 Base

6.110.1 Detailed Description

Definition at line 382 of file Module.h.

6.110.2 Member Typedef Documentation

6.110.2.1 Base

```
typedef S4 Rcpp::CppClass::Base  [private]
```

Definition at line 383 of file Module.h.

6.110.2.2 XP

```
typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppClass::XP
```

Definition at line 386 of file Module.h.

6.110.2.3 XP_Class

```
typedef XPtr<class_Base> Rcpp::CppClass::XP_Class
```

Definition at line 385 of file Module.h.

6.110.3 Constructor & Destructor Documentation

6.110.3.1 CppClass() [1/2]

```
Rcpp::CppClass::CppClass (
             SEXP x )  [inline]
```

Definition at line 387 of file Module.h.

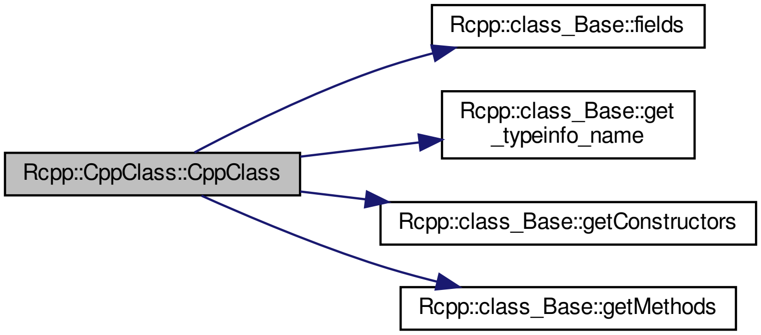
6.110.3.2 CppClass() [2/2]

```
Rcpp::CppClass::CppClass (
             Module * p,
             class_Base * cl,
             std::string & buffer )  [inline]
```

Definition at line 389 of file Module.h.

References Rcpp::class_Base::docstring, Rcpp::class_Base::enums, Rcpp::class_Base::fields(), Rcpp::class_Base↩::get_typeinfo_name(), Rcpp::class_Base::getConstructors(), Rcpp::class_Base::getMethods(), Rcpp::class_Base↩::name, and Rcpp::class_Base::parents.

Here is the call graph for this function:

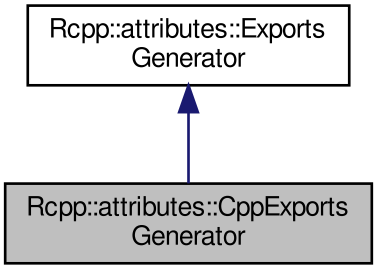

Rcpp::CppClass::CppClass
Rcpp::class_Base::fields
Rcpp::class_Base::get _typeinfo_name
Rcpp::class_Base::getConstructors
Rcpp::class_Base::getMethods

The documentation for this class was generated from the following file:

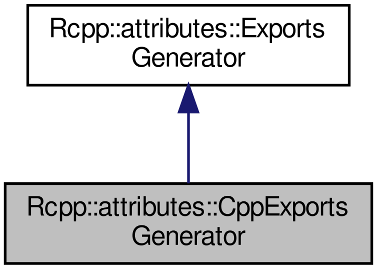
- inst/include/Rcpp/Module.h

6.111 Rcpp::attributes::CppExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppExportsGenerator:


Rcpp::attributes::Exports Generator
Rcpp::attributes::CppExports Generator

Collaboration diagram for Rcpp::attributes::CppExportsGenerator:


Rcpp::attributes::Exports Generator
Rcpp::attributes::CppExports Generator

Public Member Functions

- CppExportsGenerator (const std::string &packageDir, const std::string &package, const std::string &fileSep)
- virtual void writeBegin ()
- virtual void writeEnd (bool hasPackageInit)
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)
- std::string registerCCallable (size_t indent, const std::string &exportedName, const std::string &name) const

Private Attributes

- std::vector< Attribute > initFunctions_
- std::vector< Attribute > cppExports_
- std::vector< Attribute > nativeRoutines_
- std::vector< std::string > modules_

Additional Inherited Members

6.111.1 Detailed Description

Definition at line 696 of file attributes.cpp.

6.111.2 Constructor & Destructor Documentation

6.111.2.1 CppExportsGenerator()

```
Rcpp::attributes::CppExportsGenerator::CppExportsGenerator (
            const std::string & packageDir,
            const std::string & package,
            const std::string & fileSep )  [explicit]
```

Definition at line 1956 of file attributes.cpp.

6.111.3 Member Function Documentation

6.111.3.1 commit()

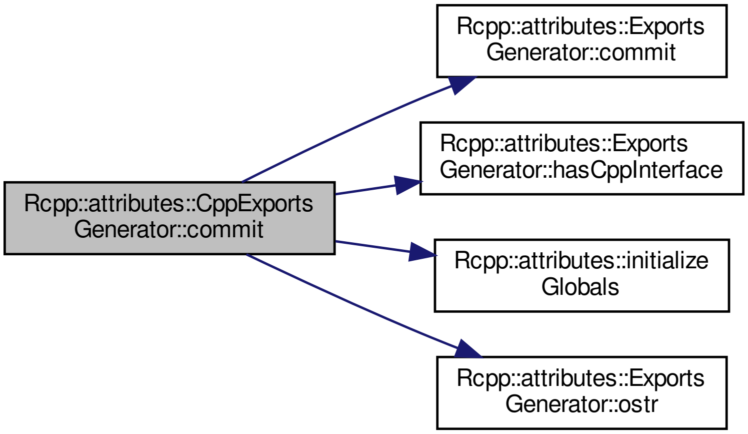
```
bool Rcpp::attributes::CppExportsGenerator::commit (
            const std::vector< std::string > & includes )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2163 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), Rcpp::attributes::initializeGlobals(), and Rcpp::attributes::ExportsGenerator::ostr().

Here is the call graph for this function:


Rcpp::attributes::CppExports Generator::commit
Rcpp::attributes::Exports Generator::commit
Rcpp::attributes::Exports Generator::hasCppInterface
Rcpp::attributes::initialize Globals
Rcpp::attributes::Exports Generator::ostr

6.111.3.2 doWriteFunctions()

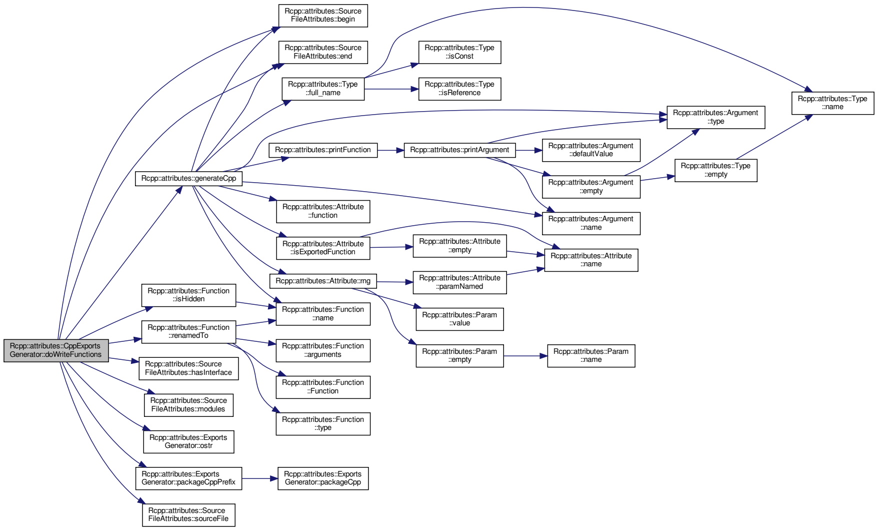
```
void Rcpp::attributes::CppExportsGenerator::doWriteFunctions (
            const SourceFileAttributes & attributes,
            bool verbose )  [private], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 1966 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), cppExports_, Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::generateCpp(), Rcpp::attributes::SourceFileAttributes::hasInterface(), initFunctions_, Rcpp::attributes::Function::isHidden(), Rcpp::attributes::kInitAttribute, Rcpp::attributes::kInterfaceCpp, Rcpp::attributes::SourceFileAttributes::modules(), modules_, nativeRoutines_, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::packageCppPrefix(), Rcpp::Rcout, Rcpp::attributes::Function::renamedTo(), and Rcpp::attributes::SourceFileAttributes::sourceFile().

Here is the call graph for this function:


Rcpp::attributes::CppExports Generator::doWriteFunctions
Rcpp::attributes::Source FileAttributes::begin
Rcpp::attributes::Source FileAttributes::end
Rcpp::attributes::Type ::full_name
Rcpp::attributes::Type ::isConst
Rcpp::attributes::Type ::isReference
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::type
Rcpp::attributes::printFunction
Rcpp::attributes::printArgument
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::Argument ::empty
Rcpp::attributes::Type ::empty
Rcpp::attributes::generateCpp
Rcpp::attributes::Argument ::name
Rcpp::attributes::Attribute ::function
Rcpp::attributes::Attribute ::isExportedFunction
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::name
Rcpp::attributes::Attribute::rng
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Function ::isHidden
Rcpp::attributes::Function ::name
Rcpp::attributes::Param ::value
Rcpp::attributes::Function ::renamedTo
Rcpp::attributes::Function ::arguments
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Source FileAttributes::hasInterface
Rcpp::attributes::Function ::Function
Rcpp::attributes::Source FileAttributes::modules
Rcpp::attributes::Function ::type
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::Source FileAttributes::sourceFile

6.111.3.3 registerCCallable()

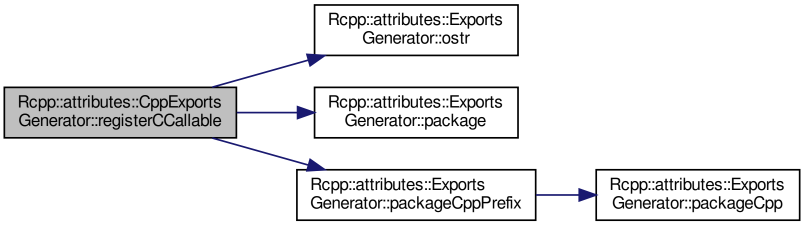
```
std::string Rcpp::attributes::CppExportsGenerator::registerCCallable (
            size_t indent,
            const std::string & exportedName,
            const std::string & name ) const  [private]
```

Definition at line 2151 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), and Rcpp::attributes::ExportsGenerator::packageCppPrefix().

Referenced by writeEnd().

Here is the call graph for this function:


Rcpp::attributes::CppExports Generator::registerCCallable
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::Exports Generator::package
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp

6.111.3.4 writeBegin()

```
virtual void Rcpp::attributes::CppExportsGenerator::writeBegin ( )  [inline], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 702 of file attributes.cpp.

6.111.3.5 writeEnd()

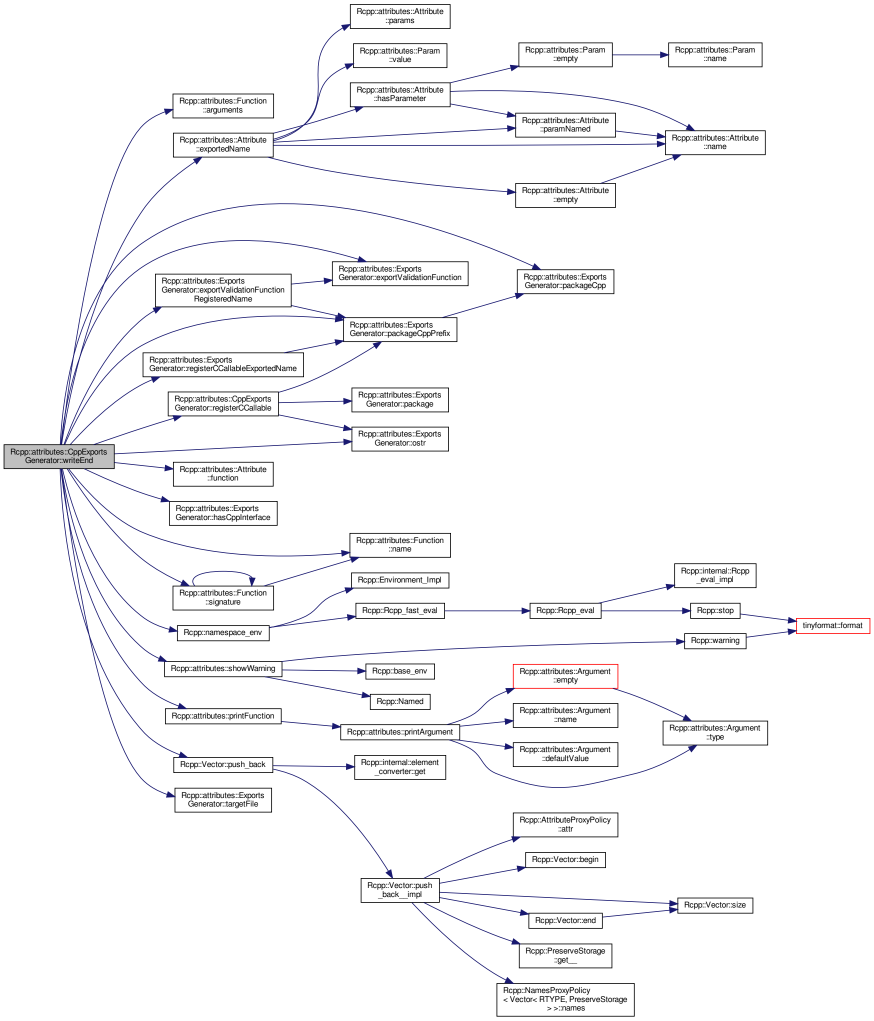
```
void Rcpp::attributes::CppExportsGenerator::writeEnd (
             bool hasPackageInit )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2018 of file attributes.cpp.

References Rcpp::attributes::Function::arguments(), callEntries, cppExports_, Rcpp::attributes::Attribute::exportedName(), Rcpp::attributes::ExportsGenerator::exportValidationFunction(), Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), initFunctions_, modules_, Rcpp::attributes::Function::name(), Rcpp::namespace_env(), nativeRoutines_, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::packageCpp(), Rcpp::attributes::ExportsGenerator::packageCppPrefix(), Rcpp::attributes::printFunction(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back(), registerCCallable(), Rcpp::attributes::ExportsGenerator::registerCCallableExportedName(), Rcpp::attributes::showWarning(), Rcpp::attributes::Function::signature(), and Rcpp::attributes::ExportsGenerator::targetFile().

Here is the call graph for this function:


Rcpp::attributes::CppExports
Generator::writeEnd
Rcpp::attributes::Function
::arguments
Rcpp::attributes::Attribute
::exportedName
Rcpp::attributes::Attribute
::params
Rcpp::attributes::Param
::value
Rcpp::attributes::Attribute
::hasParameter
Rcpp::attributes::Param
::empty
Rcpp::attributes::Param
::name
Rcpp::attributes::Attribute
::paramNamed
Rcpp::attributes::Attribute
::name
Rcpp::attributes::Attribute
::empty
Rcpp::attributes::Exports
Generator::exportValidationFunction
RegisteredName
Rcpp::attributes::Exports
Generator::exportValidationFunction
Rcpp::attributes::Exports
Generator::packageCpp
Rcpp::attributes::Exports
Generator::packageCppPrefix
Rcpp::attributes::Exports
Generator::registerCCallableExportedName
Rcpp::attributes::CppExports
Generator::registerCCallable
Rcpp::attributes::Exports
Generator::package
Rcpp::attributes::Exports
Generator::ostr
Rcpp::attributes::Attribute
::function
Rcpp::attributes::Exports
Generator::hasCppInterface
Rcpp::attributes::Function
::name
Rcpp::attributes::Function
::signature
Rcpp::Environment_Impl
Rcpp::namespace_env
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp
_eval_impl
Rcpp::stop
tinyformat::format
Rcpp::warning
Rcpp::attributes::showWarning
Rcpp::base_env
Rcpp::Named
Rcpp::attributes::printFunction
Rcpp::attributes::printArgument
Rcpp::attributes::Argument
::empty
Rcpp::attributes::Argument
::name
Rcpp::attributes::Argument
::defaultValue
Rcpp::attributes::Argument
::type
Rcpp::Vector::push_back
Rcpp::internal::element
_converter::get
Rcpp::attributes::Exports
Generator::targetFile
Rcpp::AttributeProxyPolicy
::attr
Rcpp::Vector::begin
Rcpp::Vector::push
_back__impl
Rcpp::Vector::size
Rcpp::Vector::end
Rcpp::PreserveStorage
::get__
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage
> >::names

6.111.4 Member Data Documentation

6.111.4.1 cppExports_

`std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::cppExports_ [private]`

Definition at line 719 of file attributes.cpp.

Referenced by doWriteFunctions(), and writeEnd().

6.111.4.2 initFunctions_

```
std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::initFunctions_  [private]
```

Definition at line 716 of file attributes.cpp.

Referenced by doWriteFunctions(), and writeEnd().

6.111.4.3 modules_

```
std::vector<std::string> Rcpp::attributes::CppExportsGenerator::modules_  [private]
```

Definition at line 725 of file attributes.cpp.

Referenced by doWriteFunctions(), and writeEnd().

6.111.4.4 nativeRoutines_

```
std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::nativeRoutines_  [private]
```

Definition at line 722 of file attributes.cpp.

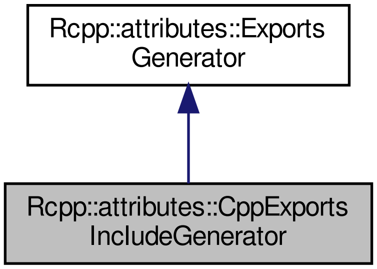
Referenced by doWriteFunctions(), and writeEnd().

The documentation for this class was generated from the following file:

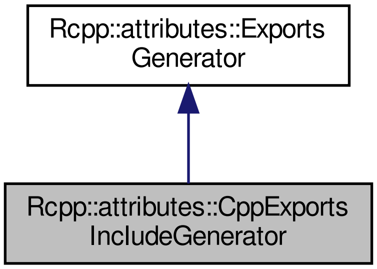
- src/attributes.cpp

6.112 Rcpp::attributes::CppExportsIncludeGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppExportsIncludeGenerator:



Collaboration diagram for Rcpp::attributes::CppExportsIncludeGenerator:



Public Member Functions

- CppExportsIncludeGenerator (const std::string &packageDir, const std::string &package, const std::string &file←Sep)
- virtual void writeBegin ()
- virtual void writeEnd (bool hasPackageInit)
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)
- std::string getCCallable (const std::string &function) const
- std::string getHeaderGuard () const

Private Attributes

- std::string includeDir_

Additional Inherited Members

6.112.1 Detailed Description

Definition at line 729 of file attributes.cpp.

6.112.2 Constructor & Destructor Documentation

6.112.2.1 CppExportsIncludeGenerator()

```
Rcpp::attributes::CppExportsIncludeGenerator::CppExportsIncludeGenerator (
             const std::string & packageDir,
             const std::string & package,
             const std::string & fileSep )
```

Definition at line 2186 of file attributes.cpp.

References includeDir_.

6.112.3 Member Function Documentation

6.112.3.1 commit()

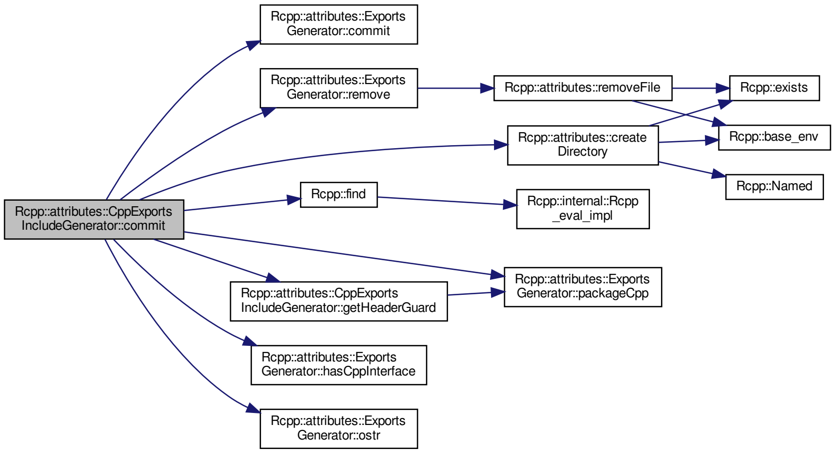
```
bool Rcpp::attributes::CppExportsIncludeGenerator::commit (
             const std::vector< std::string > & includes ) [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2339 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit(), Rcpp::attributes::createDirectory(), Rcpp::find(), get↩HeaderGuard(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), includeDir_, Rcpp::attributes::Exports↩Generator::ostr(), Rcpp::attributes::ExportsGenerator::packageCpp(), and Rcpp::attributes::ExportsGenerator↩::remove().

Here is the call graph for this function:


Rcpp::attributes::CppExports IncludeGenerator::commit
Rcpp::attributes::Exports Generator::commit
Rcpp::attributes::Exports Generator::remove
Rcpp::attributes::removeFile
Rcpp::exists
Rcpp::base_env
Rcpp::attributes::create Directory
Rcpp::Named
Rcpp::find
Rcpp::internal::Rcpp _eval_impl
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::CppExports IncludeGenerator::getHeaderGuard
Rcpp::attributes::Exports Generator::hasCppInterface
Rcpp::attributes::Exports Generator::ostr

6.112.3.2 doWriteFunctions()

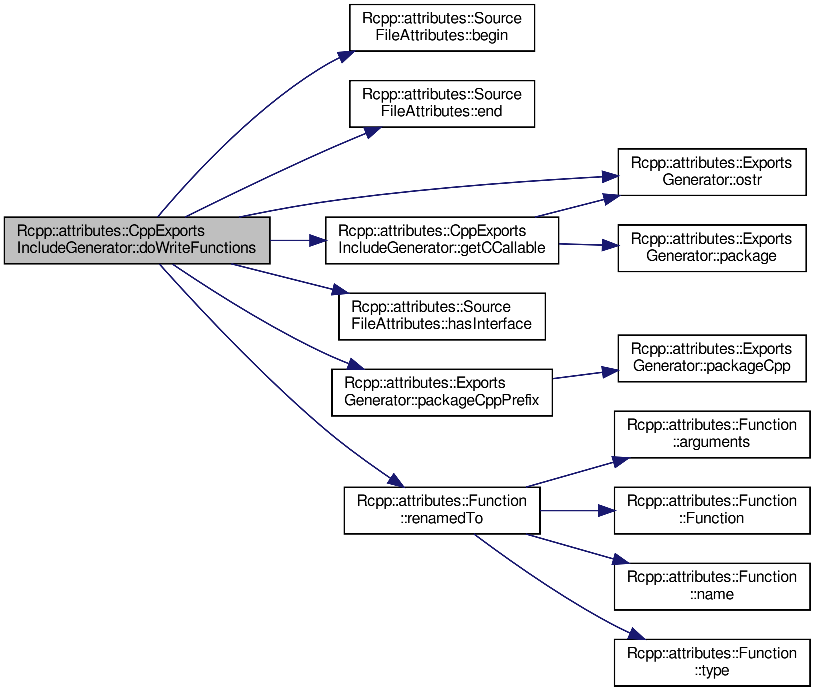
```
void Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions (
             const SourceFileAttributes & attributes,
             bool verbose )  [private], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2248 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes::end(), getCCallable(), Rcpp::attributes::SourceFileAttributes::hasInterface(), Rcpp::attributes::kInterfaceCpp, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::packageCppPrefix(), and Rcpp::attributes::Function::renamedTo().

Here is the call graph for this function:


Rcpp::attributes::CppExports IncludeGenerator::doWriteFunctions
Rcpp::attributes::Source FileAttributes::begin
Rcpp::attributes::Source FileAttributes::end
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::CppExports IncludeGenerator::getCCallable
Rcpp::attributes::Exports Generator::package
Rcpp::attributes::Source FileAttributes::hasInterface
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Function ::arguments
Rcpp::attributes::Function ::renamedTo
Rcpp::attributes::Function ::Function
Rcpp::attributes::Function ::name
Rcpp::attributes::Function ::type

6.112.3.3 getCCallable()

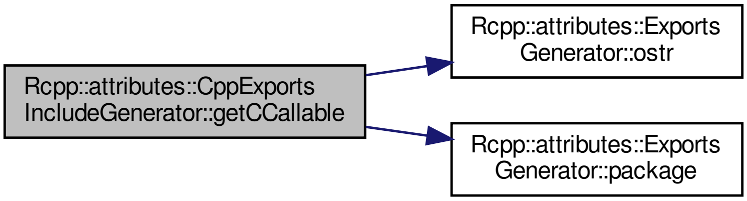
```
std::string Rcpp::attributes::CppExportsIncludeGenerator::getCCallable (
             const std::string & function ) const  [private]
```

Definition at line 2391 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::ostr(), and Rcpp::attributes::ExportsGenerator::package().

Referenced by doWriteFunctions(), and writeBegin().

Here is the call graph for this function:


Rcpp::attributes::CppExports IncludeGenerator::getCCallable
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::Exports Generator::package

6.112.3.4 getHeaderGuard()

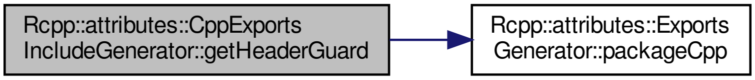
```
std::string Rcpp::attributes::CppExportsIncludeGenerator::getHeaderGuard ( ) const  [private]
```

Definition at line 2400 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::packageCpp().

Referenced by commit(), and writeEnd().

Here is the call graph for this function:


Rcpp::attributes::CppExports IncludeGenerator::getHeaderGuard
Rcpp::attributes::Exports Generator::packageCpp

6.112.3.5 writeBegin()

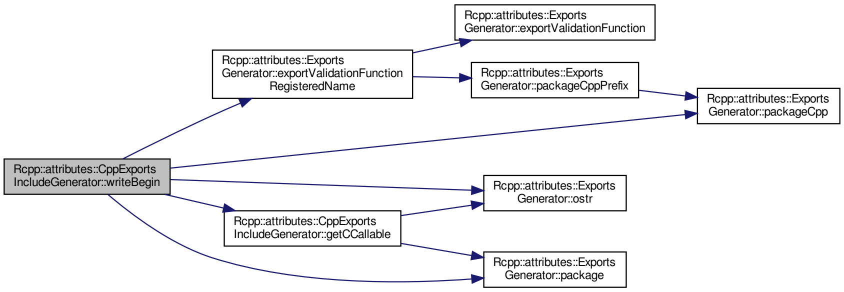
```
void Rcpp::attributes::CppExportsIncludeGenerator::writeBegin ( )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2199 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName(), getCCallable(), Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), and Rcpp::attributes::ExportsGenerator::packageCpp().

Here is the call graph for this function:


Rcpp::attributes::Exports Generator::exportValidationFunction
Rcpp::attributes::Exports Generator::exportValidationFunction RegisteredName
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::CppExports IncludeGenerator::writeBegin
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::CppExports IncludeGenerator::getCCallable
Rcpp::attributes::Exports Generator::package

6.112.3.6 writeEnd()

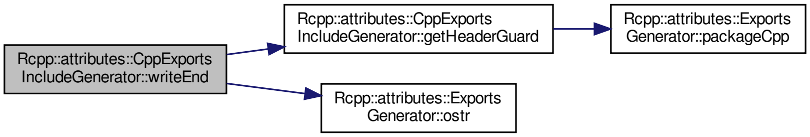
```
void Rcpp::attributes::CppExportsIncludeGenerator::writeEnd (
            bool hasPackageInit )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2333 of file attributes.cpp.

References getHeaderGuard(), and Rcpp::attributes::ExportsGenerator::ostr().

Here is the call graph for this function:


Rcpp::attributes::CppExports IncludeGenerator::getHeaderGuard
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::CppExports IncludeGenerator::writeEnd
Rcpp::attributes::Exports Generator::ostr

6.112.4 Member Data Documentation

6.112.4.1 includeDir_

```
std::string Rcpp::attributes::CppExportsIncludeGenerator::includeDir_  [private]
```

Definition at line 746 of file attributes.cpp.

Referenced by commit(), and CppExportsIncludeGenerator().

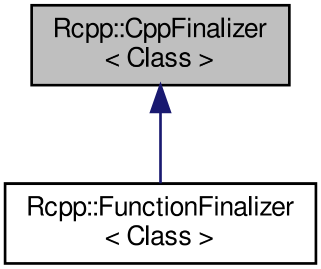
The documentation for this class was generated from the following file:

- src/attributes.cpp

6.113 Rcpp::CppFinalizer< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppFinalizer< Class >:



Public Member Functions

- CppFinalizer ()
- virtual void run (Class ∗)

6.113.1 Detailed Description

template<typename Class>
class Rcpp::CppFinalizer< Class >

Definition at line 300 of file Module.h.

6.113.2 Constructor & Destructor Documentation

6.113.2.1 CppFinalizer()

```
template<typename Class >
Rcpp::CppFinalizer< Class >::CppFinalizer ( )  [inline]
```

Definition at line 302 of file Module.h.

6.113.3 Member Function Documentation

6.113.3.1 run()

```
template<typename Class >
virtual void Rcpp::CppFinalizer< Class >::run (
             Class *  )  [inline], [virtual]
```

Reimplemented in Rcpp::FunctionFinalizer< Class >.

Definition at line 303 of file Module.h.

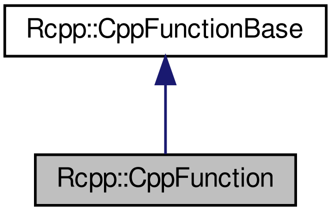
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

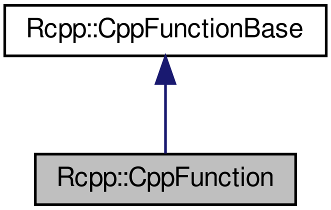
6.114 Rcpp::CppFunction Class Reference

```
#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

6.114.1 Detailed Description

base class of all exported C++ functions. Template deduction in the Module_generated_function.h file creates an instance of a class that derives CppFunction (see Module_generated_CppFunction.h for all the definitions

Definition at line 52 of file CppFunction.h.

6.114.2 Constructor & Destructor Documentation

6.114.2.1 CppFunction()

```
Rcpp::CppFunction::CppFunction (
             const char * doc = 0 )  [inline]
```

Definition at line 54 of file CppFunction.h.

6.114.2.2 ∼CppFunction()

```
virtual Rcpp::CppFunction::∼CppFunction ( )  [inline], [virtual]
```

Definition at line 55 of file CppFunction.h.

6.114.3 Member Function Documentation

6.114.3.1 get_formals()

```
virtual SEXP Rcpp::CppFunction::get_formals ( )  [inline], [virtual]
```

formal arguments

Definition at line 75 of file CppFunction.h.

Referenced by Rcpp::Module::get_function().

6.114.3.2 get_function_ptr()

```
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().

6.114.3.3 is_void()

```
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().

6.114.3.4 nargs()

```
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().

6.114.3.5 signature()

```
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().

6.114.4 Member Data Documentation

6.114.4.1 docstring

```
std::string Rcpp::CppFunction::docstring
```

description of the function

Definition at line 86 of file CppFunction.h.

Referenced by Rcpp::Module::get_function().

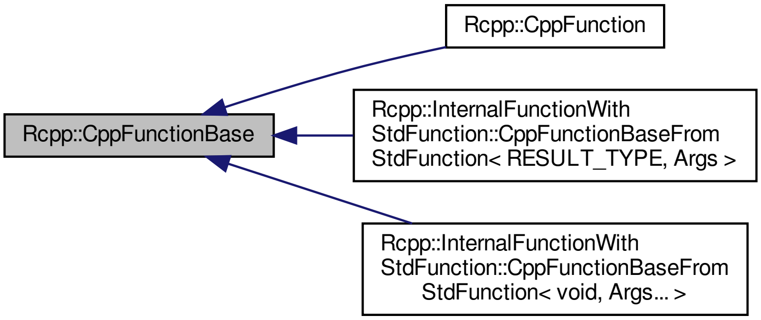
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/CppFunction.h

6.115 Rcpp::CppFunctionBase Class Reference

```
#include <CppFunction.h>
```

Inheritance diagram for Rcpp::CppFunctionBase:


Rcpp::CppFunction
Rcpp::CppFunctionBase
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< RESULT_TYPE, Args >
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< void, Args... >

Public Member Functions

- CppFunctionBase ()
- virtual ∼CppFunctionBase ()
- virtual SEXP operator() (SEXP ∗)

6.115.1 Detailed Description

base class for a callable function. This limited functionality is just barely enough for an InternalFunction, nothing more.

Definition at line 31 of file CppFunction.h.

6.115.2 Constructor & Destructor Documentation

6.115.2.1 CppFunctionBase()

```
Rcpp::CppFunctionBase::CppFunctionBase ( )  [inline]
```

Definition at line 33 of file CppFunction.h.

6.115.2.2 ∼CppFunctionBase()

```
virtual Rcpp::CppFunctionBase::∼CppFunctionBase ( )  [inline], [virtual]
```

Definition at line 34 of file CppFunction.h.

6.115.3 Member Function Documentation

6.115.3.1 operator()()

```
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.

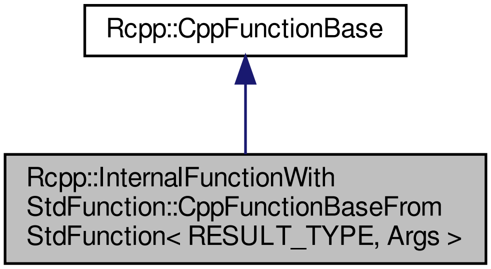
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/CppFunction.h

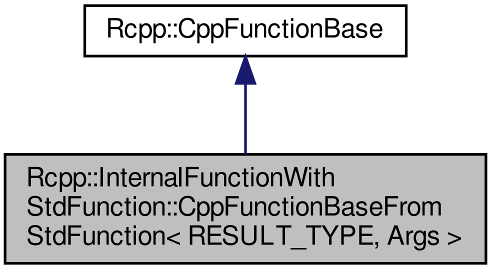
6.116 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args > Class Template Reference

```
#include <InternalFunctionWithStdFunction.h>
```

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >:


Rcpp::CppFunctionBase
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< RESULT_TYPE, Args >

Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >:


Rcpp::CppFunctionBase
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< 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

6.116.1 Detailed Description

template<typename RESULT_TYPE, typename... Args>
class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >

Definition at line 35 of file InternalFunctionWithStdFunction.h.

6.116.2 Constructor & Destructor Documentation

6.116.2.1 CppFunctionBaseFromStdFunction()

```
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.

6.116.2.2 ∼CppFunctionBaseFromStdFunction()

```
template<typename RESULT_TYPE , typename...  Args>
virtual Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args
>::∼CppFunctionBaseFromStdFunction ( )  [inline], [virtual]
```

Definition at line 38 of file InternalFunctionWithStdFunction.h.

6.116.3 Member Function Documentation

6.116.3.1 operator()()

```
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.

6.116.4 Member Data Documentation

6.116.4.1 fun

```
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()().

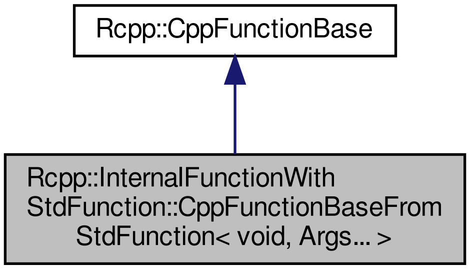
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InternalFunctionWithStdFunction.h

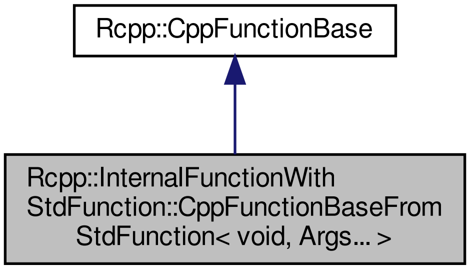
6.117 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFrom↩ StdFunction< void, Args... > Class Template Reference

```
#include <InternalFunctionWithStdFunction.h>
```

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >:


Rcpp::CppFunctionBase
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< void, Args... >

Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >:


Rcpp::CppFunctionBase
Rcpp::InternalFunctionWith StdFunction::CppFunctionBaseFrom StdFunction< 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

6.117.1 Detailed Description

template<typename... Args>
class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >

Definition at line 52 of file InternalFunctionWithStdFunction.h.

6.117.2 Constructor & Destructor Documentation

6.117.2.1 CppFunctionBaseFromStdFunction()

```
template<typename...  Args>
Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args...  >::CppFunctionBaseFromStdFu
(
            const std::function< void(Args...)> & fun )  [inline]
```

Definition at line 54 of file InternalFunctionWithStdFunction.h.

6.117.2.2 ∼CppFunctionBaseFromStdFunction()

```
template<typename...  Args>
virtual Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args...  >↩
::∼CppFunctionBaseFromStdFunction ( )  [inline], [virtual]
```

Definition at line 55 of file InternalFunctionWithStdFunction.h.

6.117.3 Member Function Documentation

6.117.3.1 operator()()

```
template<typename... Args>
SEXP Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< 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::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >::fun.

6.117.4 Member Data Documentation

6.117.4.1 fun

```
template<typename... Args>
const std::function<void(Args...)> Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction<
void, Args... >::fun [private]
```

Definition at line 64 of file InternalFunctionWithStdFunction.h.

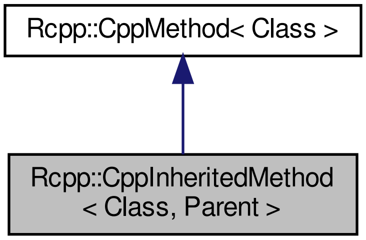
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InternalFunctionWithStdFunction.h

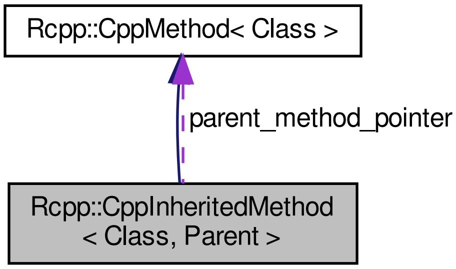
6.118 Rcpp::CppInheritedMethod< Class, Parent > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppInheritedMethod< Class, Parent >:



Collaboration diagram for Rcpp::CppInheritedMethod< Class, Parent >:



Public Types

- typedef Rcpp::XPtr< Class > XP
- typedef CppMethod< Parent > ParentMethod

Public Member Functions

- CppInheritedMethod (ParentMethod ∗parent_method_pointer_)
- SEXP operator() (Class ∗object, SEXP ∗args)
- int nargs ()
- bool is_void ()
- bool is_const ()
- void signature (std::string &s, const char ∗name)

Private Attributes

- ParentMethod ∗ parent_method_pointer

6.118.1 Detailed Description

template<typename Class, typename Parent>
class Rcpp::CppInheritedMethod< Class, Parent >

Definition at line 112 of file Module.h.

6.118.2 Member Typedef Documentation

6.118.2.1 ParentMethod

```
template<typename Class , typename Parent >
typedef CppMethod<Parent> Rcpp::CppInheritedMethod< Class, Parent >::ParentMethod
```

Definition at line 115 of file Module.h.

6.118.2.2 XP

```
template<typename Class , typename Parent >
typedef Rcpp::XPtr<Class> Rcpp::CppInheritedMethod< Class, Parent >::XP
```

Definition at line 114 of file Module.h.

6.118.3 Constructor & Destructor Documentation

6.118.3.1 CppInheritedMethod()

```
template<typename Class , typename Parent >
Rcpp::CppInheritedMethod< Class, Parent >::CppInheritedMethod (
             ParentMethod * parent_method_pointer_ )  [inline]
```

Definition at line 117 of file Module.h.

6.118.4 Member Function Documentation

6.118.4.1 is_const()

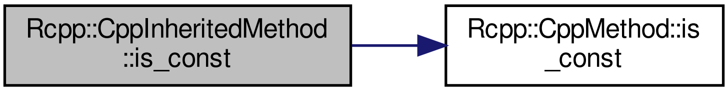
```
template<typename Class , typename Parent >
bool Rcpp::CppInheritedMethod< Class, Parent >::is_const ( )  [inline], [virtual]
```

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 126 of file Module.h.

References Rcpp::CppMethod< Class >::is_const(), and Rcpp::CppInheritedMethod< Class, Parent >::parent_↩ method_pointer.

Here is the call graph for this function:


Rcpp::CppInheritedMethod ::is_const
Rcpp::CppMethod::is _const

6.118.4.2 is_void()

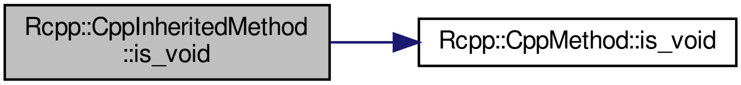
```
template<typename Class , typename Parent >
bool Rcpp::CppInheritedMethod< Class, Parent >::is_void ( )  [inline], [virtual]
```

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 125 of file Module.h.

References Rcpp::CppMethod< Class >::is_void(), and Rcpp::CppInheritedMethod< Class, Parent >::parent_↩ method_pointer.

Here is the call graph for this function:



6.118.4.3 nargs()

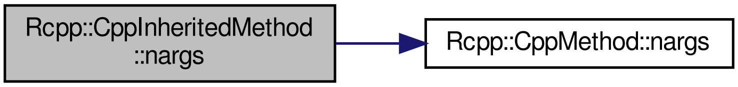
```
template<typename Class , typename Parent >
int Rcpp::CppInheritedMethod< Class, Parent >::nargs ( )  [inline], [virtual]
```

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 124 of file Module.h.

References Rcpp::CppMethod< Class >::nargs(), and Rcpp::CppInheritedMethod< Class, Parent >::parent_method_pointer.

Here is the call graph for this function:



6.118.4.4 operator()()

```
template<typename Class , typename Parent >
SEXP Rcpp::CppInheritedMethod< Class, Parent >::operator() (
             Class * object,
             SEXP * args )  [inline], [virtual]
```

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 121 of file Module.h.

References Rcpp::CppInheritedMethod< Class, Parent >::parent_method_pointer.

6.118.4.5 signature()

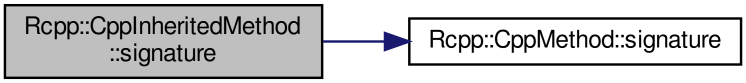
```
template<typename Class , typename Parent >
void Rcpp::CppInheritedMethod< Class, Parent >::signature (
             std::string & s,
             const char * name )  [inline], [virtual]
```

Reimplemented from Rcpp::CppMethod< Class >.

Definition at line 127 of file Module.h.

References Rcpp::CppInheritedMethod< Class, Parent >::parent_method_pointer, and Rcpp::CppMethod< Class >↩
::signature().

Here is the call graph for this function:



6.118.5 Member Data Documentation

6.118.5.1 parent_method_pointer

```
template<typename Class , typename Parent >
ParentMethod* Rcpp::CppInheritedMethod< Class, Parent >::parent_method_pointer  [private]
```

Definition at line 130 of file Module.h.

Referenced by Rcpp::CppInheritedMethod< Class, Parent >::is_const(), Rcpp::CppInheritedMethod< Class, Parent >::is_void(), Rcpp::CppInheritedMethod< Class, Parent >::nargs(), Rcpp::CppInheritedMethod< Class, Parent >↩
::operator()(), and Rcpp::CppInheritedMethod< Class, Parent >::signature().

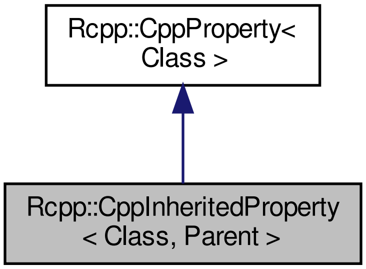
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

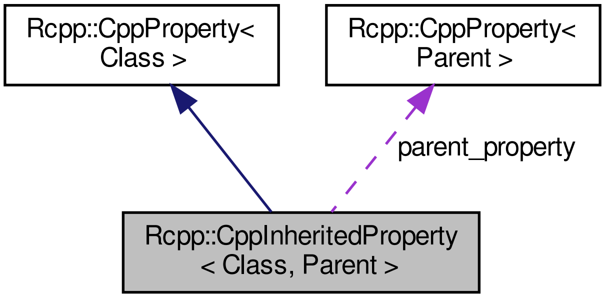
6.119 Rcpp::CppInheritedProperty< Class, Parent > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppInheritedProperty< Class, Parent >:



Collaboration diagram for Rcpp::CppInheritedProperty< Class, Parent >:



Public Types

- typedef CppProperty< Class > Base

Public Member Functions

- CppInheritedProperty (CppProperty< Parent > ∗parent_property_)
- SEXP get (Class ∗obj)
- void set (Class ∗obj, SEXP s)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- CppProperty< Parent > ∗ parent_property

Additional Inherited Members

6.119.1 Detailed Description

template<typename Class, typename Parent>
class Rcpp::CppInheritedProperty< Class, Parent >

Definition at line 281 of file Module.h.

6.119.2 Member Typedef Documentation

6.119.2.1 Base

```
template<typename Class , typename Parent >
typedef CppProperty<Class> Rcpp::CppInheritedProperty< Class, Parent >::Base
```

Definition at line 283 of file Module.h.

6.119.3 Constructor & Destructor Documentation

6.119.3.1 CppInheritedProperty()

```
template<typename Class , typename Parent >
Rcpp::CppInheritedProperty< Class, Parent >::CppInheritedProperty (
             CppProperty< Parent > ∗ parent_property_ )  [inline]
```

Definition at line 285 of file Module.h.

6.119.4 Member Function Documentation

6.119.4.1 get()

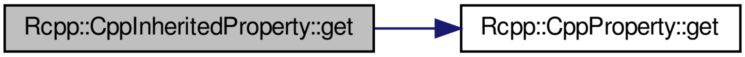
```
template<typename Class , typename Parent >
SEXP Rcpp::CppInheritedProperty< Class, Parent >::get (
            Class * obj )  [inline], [virtual]
```

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 290 of file Module.h.

References Rcpp::CppProperty< Class >::get(), and Rcpp::CppInheritedProperty< Class, Parent >::parent_property.

Here is the call graph for this function:



6.119.4.2 get_class()

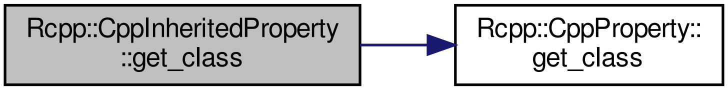
```
template<typename Class , typename Parent >
std::string Rcpp::CppInheritedProperty< Class, Parent >::get_class ( )  [inline], [virtual]
```

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 293 of file Module.h.

References Rcpp::CppProperty< Class >::get_class(), and Rcpp::CppInheritedProperty< Class, Parent >::parent_property.

Here is the call graph for this function:



6.119.4.3 is_readonly()

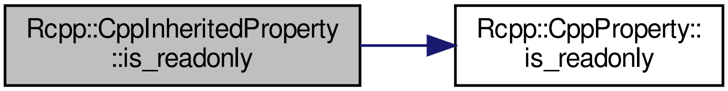
```
template<typename Class , typename Parent >
bool Rcpp::CppInheritedProperty< Class, Parent >::is_readonly ( )  [inline], [virtual]
```

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 292 of file Module.h.

References Rcpp::CppProperty< Class >::is_readonly(), and Rcpp::CppInheritedProperty< Class, Parent >::parent_property.

Here is the call graph for this function:


Rcpp::CppInheritedProperty ::is_readonly
Rcpp::CppProperty:: is_readonly

6.119.4.4 set()

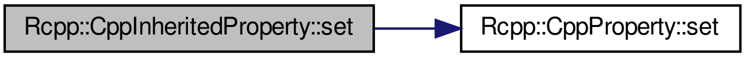
```
template<typename Class , typename Parent >
void Rcpp::CppInheritedProperty< Class, Parent >::set (
            Class * obj,
            SEXP s )  [inline], [virtual]
```

Reimplemented from Rcpp::CppProperty< Class >.

Definition at line 291 of file Module.h.

References Rcpp::CppInheritedProperty< Class, Parent >::parent_property, and Rcpp::CppProperty< Class >::set().

Here is the call graph for this function:



6.119.5 Member Data Documentation

6.119.5.1 parent_property

```
template<typename Class , typename Parent >
CppProperty<Parent>* Rcpp::CppInheritedProperty< Class, Parent >::parent_property  [private]
```

Definition at line 296 of file Module.h.

Referenced by Rcpp::CppInheritedProperty< Class, Parent >::get(), Rcpp::CppInheritedProperty< Class, Parent >↩ ::get_class(), Rcpp::CppInheritedProperty< Class, Parent >::is_readonly(), and Rcpp::CppInheritedProperty< Class, Parent >::set().

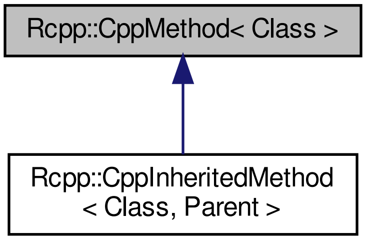
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.120 Rcpp::CppMethod< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppMethod< Class >:



Public Types

- typedef Rcpp::XPtr< Class > XP

Public Member Functions

- CppMethod ()
- virtual SEXP operator() (Class ∗, SEXP ∗)
- virtual ∼CppMethod ()
- virtual int nargs ()
- virtual bool is_void ()
- virtual bool is_const ()
- virtual void signature (std::string &s, const char ∗name)

6.120.1 Detailed Description

template<typename Class>
class Rcpp::CppMethod< Class >

Definition at line 98 of file Module.h.

6.120.2 Member Typedef Documentation

6.120.2.1 XP

```
template<typename Class >
typedef Rcpp::XPtr<Class> Rcpp::CppMethod< Class >::XP
```

Definition at line 100 of file Module.h.

6.120.3 Constructor & Destructor Documentation

6.120.3.1 CppMethod()

```
template<typename Class >
Rcpp::CppMethod< Class >::CppMethod ( )  [inline]
```

Definition at line 102 of file Module.h.

6.120.3.2 ∼CppMethod()

```
template<typename Class >
virtual Rcpp::CppMethod< Class >::∼CppMethod ( )  [inline], [virtual]
```

Definition at line 104 of file Module.h.

6.120.4 Member Function Documentation

6.120.4.1 is_const()

```
template<typename Class >
virtual bool Rcpp::CppMethod< Class >::is_const ( )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 107 of file Module.h.

Referenced by Rcpp::CppInheritedMethod< Class, Parent >::is_const(), and Rcpp::SignedMethod< Class >::is_const().

6.120.4.2 is_void()

```
template<typename Class >
virtual bool Rcpp::CppMethod< Class >::is_void ( )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 106 of file Module.h.

Referenced by Rcpp::CppInheritedMethod< Class, Parent >::is_void(), and Rcpp::SignedMethod< Class >::is_void().

6.120.4.3 nargs()

```
template<typename Class >
virtual int Rcpp::CppMethod< Class >::nargs ( )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 105 of file Module.h.

Referenced by Rcpp::CppInheritedMethod< Class, Parent >::nargs(), and Rcpp::SignedMethod< Class >::nargs().

6.120.4.4 operator()()

```
template<typename Class >
virtual SEXP Rcpp::CppMethod< Class >::operator() (
            Class * ,
            SEXP *  )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 103 of file Module.h.

6.120.4.5 signature()

```
template<typename Class >
virtual void Rcpp::CppMethod< Class >::signature (
            std::string & s,
            const char * name )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedMethod< Class, Parent >.

Definition at line 108 of file Module.h.

Referenced by Rcpp::CppInheritedMethod< Class, Parent >::signature(), and Rcpp::SignedMethod< Class >↩ ::signature().

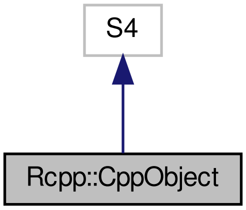
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

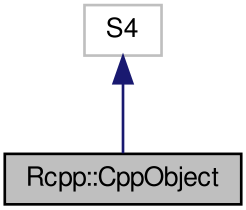
6.121 Rcpp::CppObject Class Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppObject:



Collaboration diagram for Rcpp::CppObject:



Public Types

- typedef Rcpp::XPtr< Rcpp::Module > XP

Public Member Functions

- CppObject (Module ∗p, class_Base ∗clazz, SEXP xp)

Private Types

- typedef S4 Base

6.121.1 Detailed Description

Definition at line 412 of file Module.h.

6.121.2 Member Typedef Documentation

6.121.2.1 Base

```
typedef S4 Rcpp::CppObject::Base  [private]
```

Definition at line 413 of file Module.h.

6.121.2.2 XP

```
typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppObject::XP
```

Definition at line 415 of file Module.h.

6.121.3 Constructor & Destructor Documentation

6.121.3.1 CppObject()

```
Rcpp::CppObject::CppObject (
            Module * p,
            class_Base * clazz,
            SEXP xp )  [inline]
```

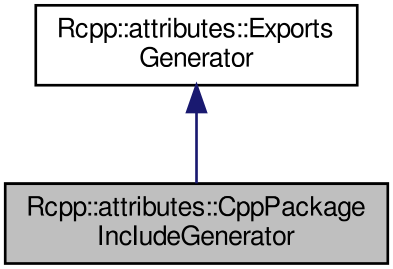
Definition at line 416 of file Module.h.

The documentation for this class was generated from the following file:

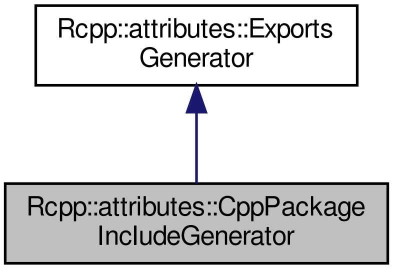
- inst/include/Rcpp/Module.h

6.122 Rcpp::attributes::CppPackageIncludeGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppPackageIncludeGenerator:


Rcpp::attributes::Exports Generator
Rcpp::attributes::CppPackage IncludeGenerator

Collaboration diagram for Rcpp::attributes::CppPackageIncludeGenerator:


Rcpp::attributes::Exports Generator
Rcpp::attributes::CppPackage IncludeGenerator

Public Member Functions

- CppPackageIncludeGenerator (const std::string &packageDir, const std::string &package, const std::string &file←Sep)
- virtual void writeBegin ()
- virtual void writeEnd (bool hasPackageInit)
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &, bool)
- std::string getHeaderGuard () const

Private Attributes

- std::string includeDir_

Additional Inherited Members

6.122.1 Detailed Description

Definition at line 750 of file attributes.cpp.

6.122.2 Constructor & Destructor Documentation

6.122.2.1 CppPackageIncludeGenerator()

```
Rcpp::attributes::CppPackageIncludeGenerator::CppPackageIncludeGenerator (
            const std::string & packageDir,
            const std::string & package,
            const std::string & fileSep )
```

Definition at line 2404 of file attributes.cpp.

References includeDir_.

6.122.3 Member Function Documentation

6.122.3.1 commit()

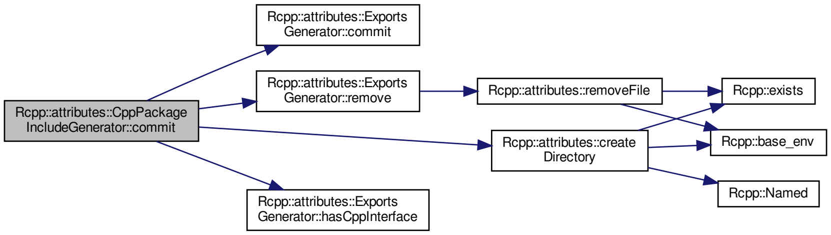
```
bool Rcpp::attributes::CppPackageIncludeGenerator::commit (
            const std::vector< std::string > & includes )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2431 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit(), Rcpp::attributes::createDirectory(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), includeDir_, and Rcpp::attributes::ExportsGenerator::remove().

Here is the call graph for this function:


Rcpp::attributes::Exports Generator::commit
Rcpp::attributes::Exports Generator::remove
Rcpp::attributes::removeFile
Rcpp::exists
Rcpp::attributes::CppPackage IncludeGenerator::commit
Rcpp::base_env
Rcpp::attributes::create Directory
Rcpp::Named
Rcpp::attributes::Exports Generator::hasCppInterface

6.122.3.2 doWriteFunctions()

```
virtual void Rcpp::attributes::CppPackageIncludeGenerator::doWriteFunctions (
             const SourceFileAttributes & ,
             bool  )  [inline], [private], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 761 of file attributes.cpp.

6.122.3.3 getHeaderGuard()

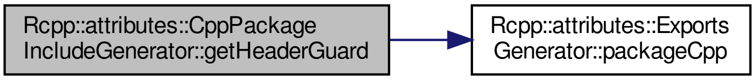
```
std::string Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard ( ) const  [private]
```

Definition at line 2445 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::packageCpp().

Referenced by writeEnd().

Here is the call graph for this function:


Rcpp::attributes::CppPackage
IncludeGenerator::getHeaderGuard
Rcpp::attributes::Exports
Generator::packageCpp

6.122.3.4 writeBegin()

```
virtual void Rcpp::attributes::CppPackageIncludeGenerator::writeBegin ( )  [inline], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 756 of file attributes.cpp.

6.122.3.5 writeEnd()

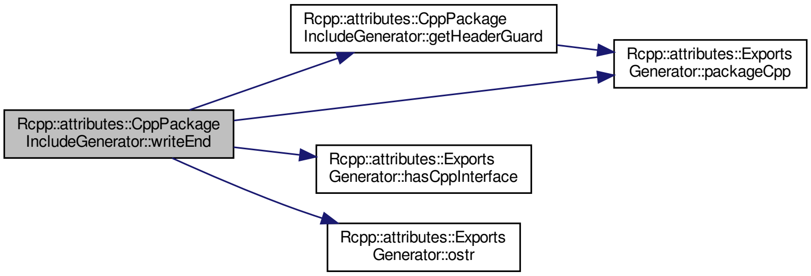
```
void Rcpp::attributes::CppPackageIncludeGenerator::writeEnd (
            bool hasPackageInit )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2417 of file attributes.cpp.

References getHeaderGuard(), Rcpp::attributes::ExportsGenerator::hasCppInterface(), Rcpp::attributes::Exports↩ Generator::ostr(), and Rcpp::attributes::ExportsGenerator::packageCpp().

Here is the call graph for this function:


Rcpp::attributes::CppPackage IncludeGenerator::getHeaderGuard
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::CppPackage IncludeGenerator::writeEnd
Rcpp::attributes::Exports Generator::hasCppInterface
Rcpp::attributes::Exports Generator::ostr

6.122.4 Member Data Documentation

6.122.4.1 includeDir_

```
std::string Rcpp::attributes::CppPackageIncludeGenerator::includeDir_  [private]
```

Definition at line 765 of file attributes.cpp.

Referenced by commit(), and CppPackageIncludeGenerator().

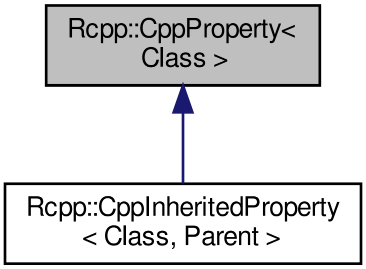
The documentation for this class was generated from the following file:

- src/attributes.cpp

6.123 Rcpp::CppProperty< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppProperty< Class >:


Rcpp::CppProperty< Class >
Rcpp::CppInheritedProperty < Class, Parent >

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

6.123.1 Detailed Description

template<typename Class>
class Rcpp::CppProperty< Class >

Definition at line 266 of file Module.h.

6.123.2 Member Typedef Documentation

6.123.2.1 XP

```
template<typename Class >
typedef Rcpp::XPtr<Class> Rcpp::CppProperty< Class >::XP
```

Definition at line 268 of file Module.h.

6.123.3 Constructor & Destructor Documentation

6.123.3.1 CppProperty()

```
template<typename Class >
Rcpp::CppProperty< Class >::CppProperty (
             const char * doc = 0 )  [inline]
```

Definition at line 270 of file Module.h.

6.123.3.2 ∼CppProperty()

```
template<typename Class >
virtual Rcpp::CppProperty< Class >::∼CppProperty ( )  [inline], [virtual]
```

Definition at line 271 of file Module.h.

6.123.4 Member Function Documentation

6.123.4.1 get()

```
template<typename Class >
virtual SEXP Rcpp::CppProperty< Class >::get (
             Class * )  [inline], [virtual]
```

Reimplemented in Rcpp::CppInheritedProperty< Class, Parent >.

Definition at line 272 of file Module.h.

Referenced by Rcpp::CppInheritedProperty< Class, Parent >::get().

6.123.4.2 get_class()

```
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::CppInheritedProperty< Class, Parent >::get_class(), and Rcpp::S4_field< Class >::S4_field().

6.123.4.3 is_readonly()

```
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::CppInheritedProperty< Class, Parent >::is_readonly(), and Rcpp::S4_field< Class >::S4_field().

6.123.4.4 set()

```
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.

Referenced by Rcpp::CppInheritedProperty< Class, Parent >::set().

6.123.5 Member Data Documentation

6.123.5.1 docstring

```
template<typename Class >
std::string Rcpp::CppProperty< Class >::docstring
```

Definition at line 277 of file Module.h.

Referenced by Rcpp::S4_field< Class >::S4_field().

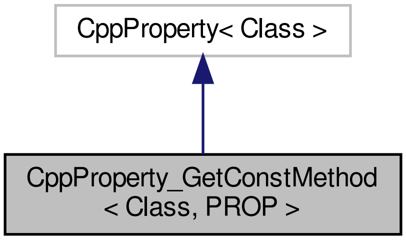
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

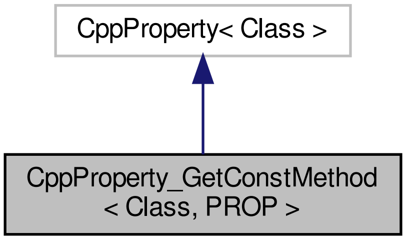
6.124 CppProperty_GetConstMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetConstMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetConstMethod< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void) const
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetConstMethod (GetMethod getter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗, SEXP)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- std::string class_name

6.124.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod< Class, PROP >

Definition at line 48 of file Module_Property.h.

6.124.2 Member Typedef Documentation

6.124.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetConstMethod< Class, PROP >::GetMethod) (void) const
```

Definition at line 50 of file Module_Property.h.

6.124.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetConstMethod< Class, PROP >::prop_class
```

Definition at line 51 of file Module_Property.h.

6.124.3 Constructor & Destructor Documentation

6.124.3.1 CppProperty_GetConstMethod()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod< Class, PROP >::CppProperty_GetConstMethod (
             GetMethod getter_,
             const char * doc = 0 )  [inline]
```

Definition at line 53 of file Module_Property.h.

6.124.4 Member Function Documentation

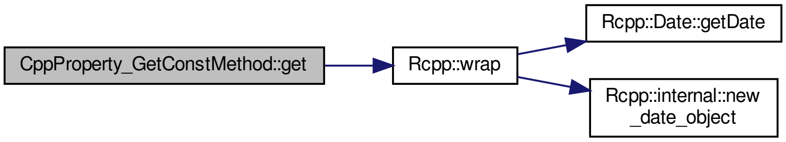
6.124.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetConstMethod< Class, PROP >::get (
             Class * object )  [inline]
```

Definition at line 56 of file Module_Property.h.

References CppProperty_GetConstMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.124.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod< Class, PROP >::get_class ( )  [inline]
```

Definition at line 59 of file Module_Property.h.

References CppProperty_GetConstMethod< Class, PROP >::class_name.

6.124.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetConstMethod< Class, PROP >::is_readonly ( )  [inline]
```

Definition at line 58 of file Module_Property.h.

6.124.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetConstMethod< Class, PROP >::set (
             Class * ,
             SEXP   )  [inline]
```

Definition at line 57 of file Module_Property.h.

6.124.5 Member Data Documentation

6.124.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod< Class, PROP >::class_name  [private]
```

Definition at line 63 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod< Class, PROP >::get_class().

6.124.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod< Class, PROP >::getter  [private]
```

Definition at line 62 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod< Class, PROP >::get().

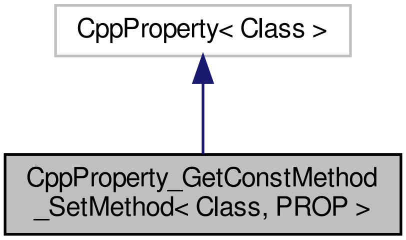
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

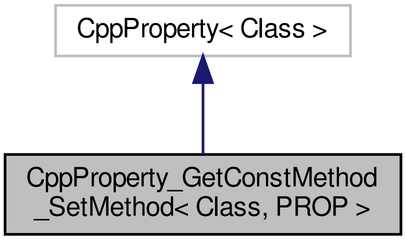
6.125 CppProperty_GetConstMethod_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetConstMethod_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetConstMethod_SetMethod< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void) const
- typedef void(Class::∗ SetMethod) (PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetConstMethod_SetMethod (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.125.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetMethod< Class, PROP >

Definition at line 115 of file Module_Property.h.

6.125.2 Member Typedef Documentation

6.125.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetConstMethod_SetMethod< Class, PROP >::GetMethod) (void)
const
```

Definition at line 117 of file Module_Property.h.

6.125.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetConstMethod_SetMethod< Class, PROP >::prop_class
```

Definition at line 119 of file Module_Property.h.

6.125.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(Class::* CppProperty_GetConstMethod_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 118 of file Module_Property.h.

6.125.3 Constructor & Destructor Documentation

6.125.3.1 CppProperty_GetConstMethod_SetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod_SetMethod< Class, PROP >::CppProperty_GetConstMethod_SetMethod (
             GetMethod getter_,
             SetMethod setter_,
             const char * doc = 0 )  [inline]
```

Definition at line 121 of file Module_Property.h.

6.125.4 Member Function Documentation

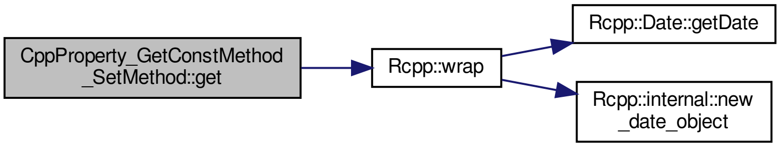
6.125.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetConstMethod_SetMethod< Class, PROP >::get (
            Class * object ) [inline]
```

Definition at line 124 of file Module_Property.h.

References CppProperty_GetConstMethod_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.125.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 131 of file Module_Property.h.

References CppProperty_GetConstMethod_SetMethod< Class, PROP >::class_name.

6.125.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetConstMethod_SetMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 130 of file Module_Property.h.

6.125.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetConstMethod_SetMethod< Class, PROP >::set (
             Class * object,
             SEXP value ) [inline]
```

Definition at line 127 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetConstMethod_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:

6.125.5 Member Data Documentation

6.125.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::class_name [private]
```

Definition at line 136 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetMethod< Class, PROP >::get_class().

6.125.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::getter [private]
```

Definition at line 134 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetMethod< Class, PROP >::get().

6.125.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::setter  [private]
```

Definition at line 135 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetMethod< Class, PROP >::set().

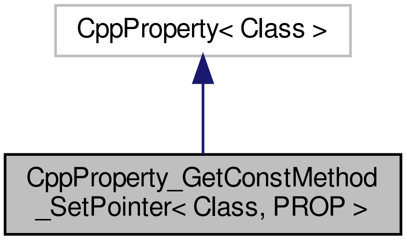
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

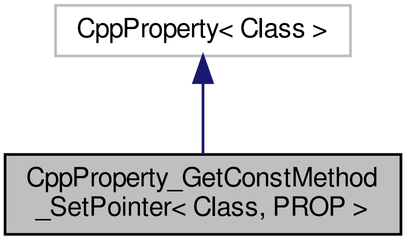
6.126 CppProperty_GetConstMethod_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetConstMethod_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty_GetConstMethod_SetPointer< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void) const
- typedef void(∗ SetMethod) (Class ∗, PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetConstMethod_SetPointer (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.126.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetPointer< Class, PROP >

Definition at line 170 of file Module_Property.h.

6.126.2 Member Typedef Documentation

6.126.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetConstMethod_SetPointer< Class, PROP >::GetMethod) (void)
const
```

Definition at line 172 of file Module_Property.h.

6.126.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetConstMethod_SetPointer< Class, PROP >::prop_class
```

Definition at line 174 of file Module_Property.h.

6.126.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetConstMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 173 of file Module_Property.h.

6.126.3 Constructor & Destructor Documentation

6.126.3.1 CppProperty_GetConstMethod_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod_SetPointer< Class, PROP >::CppProperty_GetConstMethod_SetPointer (
                GetMethod getter_,
                SetMethod setter_,
                const char * doc = 0 )  [inline]
```

Definition at line 176 of file Module_Property.h.

6.126.4 Member Function Documentation

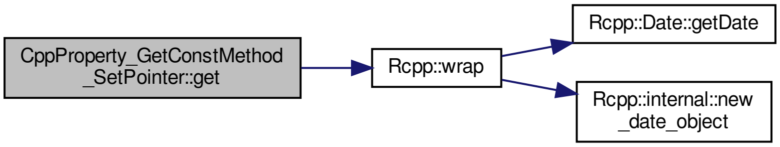
6.126.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetConstMethod_SetPointer< Class, PROP >::get (
            Class * object ) [inline]
```

Definition at line 179 of file Module_Property.h.

References CppProperty_GetConstMethod_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.126.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 186 of file Module_Property.h.

References CppProperty_GetConstMethod_SetPointer< Class, PROP >::class_name.

6.126.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetConstMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 185 of file Module_Property.h.

6.126.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetConstMethod_SetPointer< Class, PROP >::set (
             Class * object,
             SEXP value )  [inline]
```

Definition at line 182 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetConstMethod_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:

CppProperty_GetConstMethod
_SetPointer::set
Rcpp::as

6.126.5 Member Data Documentation

6.126.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::class_name  [private]
```

Definition at line 191 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetPointer< Class, PROP >::get_class().

6.126.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::getter  [private]
```

Definition at line 189 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetPointer< Class, PROP >::get().

6.126.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::setter  [private]
```

Definition at line 190 of file Module_Property.h.

Referenced by CppProperty_GetConstMethod_SetPointer< Class, PROP >::set().

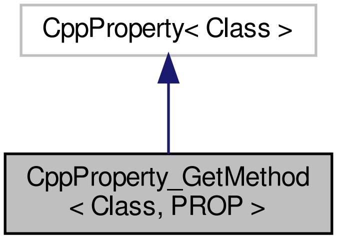
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

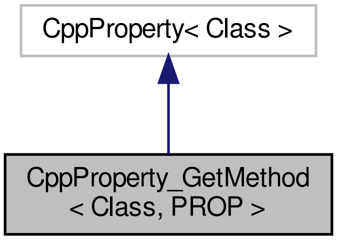
6.127 CppProperty_GetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetMethod< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetMethod (GetMethod getter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗, SEXP)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- std::string class_name

6.127.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetMethod< Class, PROP >

Definition at line 27 of file Module_Property.h.

6.127.2 Member Typedef Documentation

6.127.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetMethod< Class, PROP >::GetMethod) (void)
```

Definition at line 29 of file Module_Property.h.

6.127.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetMethod< Class, PROP >::prop_class
```

Definition at line 30 of file Module_Property.h.

6.127.3 Constructor & Destructor Documentation

6.127.3.1 CppProperty_GetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetMethod< Class, PROP >::CppProperty_GetMethod (
             GetMethod getter_,
             const char * doc = 0 )  [inline]
```

Definition at line 32 of file Module_Property.h.

6.127.4 Member Function Documentation

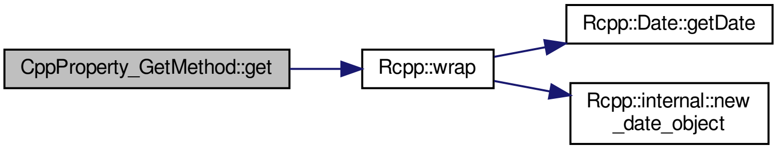
6.127.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetMethod< Class, PROP >::get (
             Class * object )  [inline]
```

Definition at line 35 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.127.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod< Class, PROP >::get_class ( )  [inline]
```

Definition at line 38 of file Module_Property.h.

References CppProperty_GetMethod< Class, PROP >::class_name.

6.127.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod< Class, PROP >::is_readonly ( )  [inline]
```

Definition at line 37 of file Module_Property.h.

6.127.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod< Class, PROP >::set (
             Class * ,
             SEXP  )  [inline]
```

Definition at line 36 of file Module_Property.h.

6.127.5 Member Data Documentation

6.127.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod< Class, PROP >::class_name  [private]
```

Definition at line 42 of file Module_Property.h.

Referenced by CppProperty_GetMethod< Class, PROP >::get_class().

6.127.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod< Class, PROP >::getter  [private]
```

Definition at line 41 of file Module_Property.h.

Referenced by CppProperty_GetMethod< Class, PROP >::get().

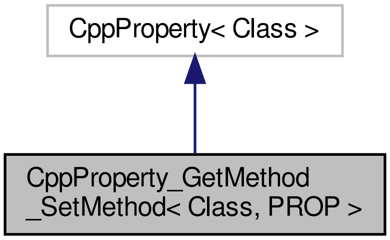
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

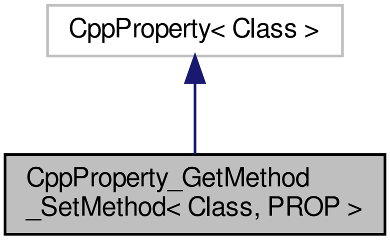
6.128 CppProperty_GetMethod_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetMethod_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetMethod_SetMethod< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void)
- typedef void(Class::∗ SetMethod) (PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetMethod_SetMethod (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.128.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetMethod_SetMethod< Class, PROP >

Definition at line 91 of file Module_Property.h.

6.128.2 Member Typedef Documentation

6.128.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetMethod_SetMethod< Class, PROP >::GetMethod) (void)
```

Definition at line 93 of file Module_Property.h.

6.128.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetMethod_SetMethod< Class, PROP >::prop_class
```

Definition at line 95 of file Module_Property.h.

6.128.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(Class::* CppProperty_GetMethod_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 94 of file Module_Property.h.

6.128.3 Constructor & Destructor Documentation

6.128.3.1 CppProperty_GetMethod_SetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetMethod_SetMethod< Class, PROP >::CppProperty_GetMethod_SetMethod (
            GetMethod getter_,
            SetMethod setter_,
            const char * doc = 0 )  [inline]
```

Definition at line 97 of file Module_Property.h.

6.128.4 Member Function Documentation

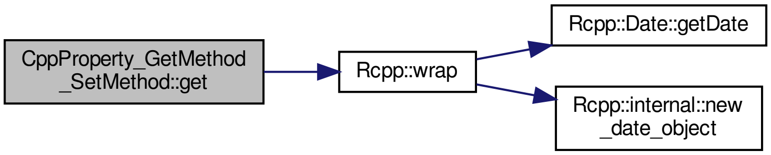
6.128.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetMethod_SetMethod< Class, PROP >::get (
            Class * object )  [inline]
```

Definition at line 100 of file Module_Property.h.

References CppProperty_GetMethod_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.128.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetMethod< Class, PROP >::get_class ( )  [inline]
```

Definition at line 107 of file Module_Property.h.

References CppProperty_GetMethod_SetMethod< Class, PROP >::class_name.

6.128.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod_SetMethod< Class, PROP >::is_readonly ( )  [inline]
```

Definition at line 106 of file Module_Property.h.

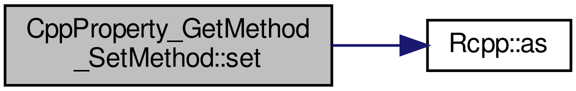
6.128.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod_SetMethod< Class, PROP >::set (
             Class * object,
             SEXP value )  [inline]
```

Definition at line 103 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetMethod_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:



6.128.5 Member Data Documentation

6.128.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetMethod< Class, PROP >::class_name  [private]
```

Definition at line 112 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetMethod< Class, PROP >::get_class().

6.128.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::getter  [private]
```

Definition at line 110 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetMethod< Class, PROP >::get().

6.128.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::setter  [private]
```

Definition at line 111 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetMethod< Class, PROP >::set().

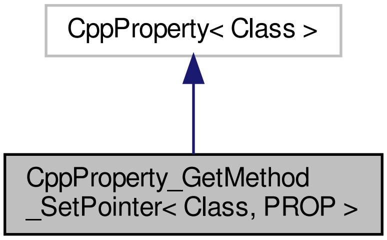
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

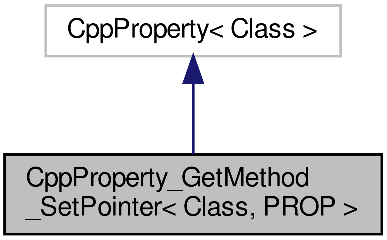
6.129 CppProperty_GetMethod_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetMethod_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty_GetMethod_SetPointer< Class, PROP >:



Public Types

- typedef PROP(Class::∗ GetMethod) (void)
- typedef void(∗ SetMethod) (Class ∗, PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetMethod_SetPointer (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.129.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetMethod_SetPointer< Class, PROP >

Definition at line 145 of file Module_Property.h.

6.129.2 Member Typedef Documentation

6.129.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::∗ CppProperty_GetMethod_SetPointer< Class, PROP >::GetMethod) (void)
```

Definition at line 147 of file Module_Property.h.

6.129.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetMethod_SetPointer< Class, PROP >::prop_class
```

Definition at line 149 of file Module_Property.h.

6.129.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 148 of file Module_Property.h.

6.129.3 Constructor & Destructor Documentation

6.129.3.1 CppProperty_GetMethod_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetMethod_SetPointer< Class, PROP >::CppProperty_GetMethod_SetPointer (
             GetMethod getter_,
             SetMethod setter_,
             const char * doc = 0 )  [inline]
```

Definition at line 151 of file Module_Property.h.

6.129.4 Member Function Documentation

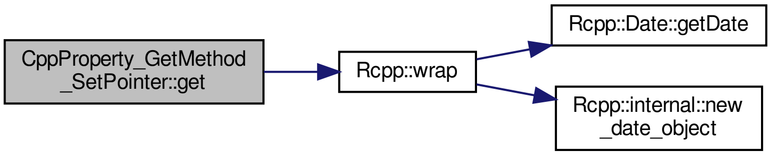
6.129.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetMethod_SetPointer< Class, PROP >::get (
            Class * object ) [inline]
```

Definition at line 154 of file Module_Property.h.

References CppProperty_GetMethod_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.129.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 161 of file Module_Property.h.

References CppProperty_GetMethod_SetPointer< Class, PROP >::class_name.

6.129.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 160 of file Module_Property.h.

6.129.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod_SetPointer< Class, PROP >::set (
             Class * object,
             SEXP value )  [inline]
```

Definition at line 157 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetMethod_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:

CppProperty_GetMethod
_SetPointer::set
Rcpp::as

6.129.5 Member Data Documentation

6.129.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetPointer< Class, PROP >::class_name  [private]
```

Definition at line 166 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetPointer< Class, PROP >::get_class().

6.129.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::getter  [private]
```

Definition at line 164 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetPointer< Class, PROP >::get().

6.129.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::setter  [private]
```

Definition at line 165 of file Module_Property.h.

Referenced by CppProperty_GetMethod_SetPointer< Class, PROP >::set().

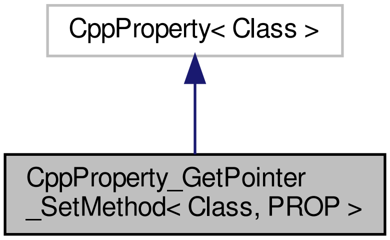
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

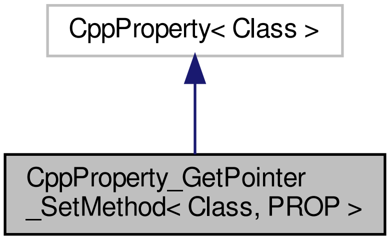
6.130 CppProperty_GetPointer_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetPointer_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetPointer_SetMethod< Class, PROP >:



Public Types

- typedef PROP(∗ GetMethod) (Class ∗)
- typedef void(Class::∗ SetMethod) (PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetPointer_SetMethod (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.130.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointer_SetMethod< Class, PROP >

Definition at line 197 of file Module_Property.h.

6.130.2 Member Typedef Documentation

6.130.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(∗ CppProperty_GetPointer_SetMethod< Class, PROP >::GetMethod) (Class ∗)
```

Definition at line 199 of file Module_Property.h.

6.130.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetPointer_SetMethod< Class, PROP >::prop_class
```

Definition at line 201 of file Module_Property.h.

6.130.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(Class::* CppProperty_GetPointer_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 200 of file Module_Property.h.

6.130.3 Constructor & Destructor Documentation

6.130.3.1 CppProperty_GetPointer_SetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetPointer_SetMethod< Class, PROP >::CppProperty_GetPointer_SetMethod (
            GetMethod getter_,
            SetMethod setter_,
            const char * doc = 0 )  [inline]
```

Definition at line 203 of file Module_Property.h.

6.130.4 Member Function Documentation

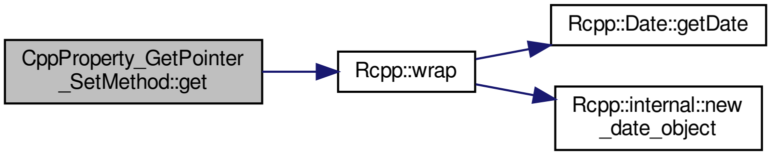
6.130.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetPointer_SetMethod< Class, PROP >::get (
            Class * object )  [inline]
```

Definition at line 206 of file Module_Property.h.

References CppProperty_GetPointer_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.130.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetMethod< Class, PROP >::get_class ( )  [inline]
```

Definition at line 213 of file Module_Property.h.

References CppProperty_GetPointer_SetMethod< Class, PROP >::class_name.

6.130.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointer_SetMethod< Class, PROP >::is_readonly ( )  [inline]
```

Definition at line 212 of file Module_Property.h.

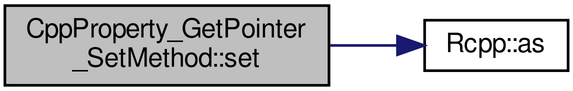
6.130.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointer_SetMethod< Class, PROP >::set (
             Class * object,
             SEXP value ) [inline]
```

Definition at line 209 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetPointer_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:



6.130.5 Member Data Documentation

6.130.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetMethod< Class, PROP >::class_name [private]
```

Definition at line 218 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetMethod< Class, PROP >::get_class().

6.130.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::getter [private]
```

Definition at line 216 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetMethod< Class, PROP >::get().

6.130.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::setter  [private]
```

Definition at line 217 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetMethod< Class, PROP >::set().

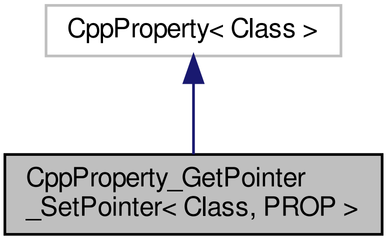
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

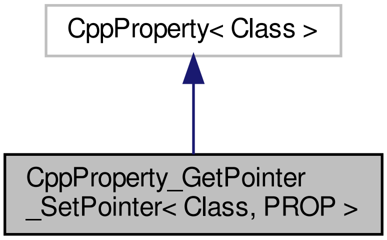
6.131 CppProperty_GetPointer_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetPointer_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty_GetPointer_SetPointer< Class, PROP >:



Public Types

- typedef PROP(∗ GetMethod) (Class ∗)
- typedef void(∗ SetMethod) (Class ∗, PROP)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetPointer_SetPointer (GetMethod getter_, SetMethod setter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- SetMethod setter
- std::string class_name

6.131.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointer_SetPointer< Class, PROP >

Definition at line 225 of file Module_Property.h.

6.131.2 Member Typedef Documentation

6.131.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(∗ CppProperty_GetPointer_SetPointer< Class, PROP >::GetMethod) (Class ∗)
```

Definition at line 227 of file Module_Property.h.

6.131.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetPointer_SetPointer< Class, PROP >::prop_class
```

Definition at line 229 of file Module_Property.h.

6.131.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetPointer_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 228 of file Module_Property.h.

6.131.3 Constructor & Destructor Documentation

6.131.3.1 CppProperty_GetPointer_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetPointer_SetPointer< Class, PROP >::CppProperty_GetPointer_SetPointer (
             GetMethod getter_,
             SetMethod setter_,
             const char * doc = 0 )  [inline]
```

Definition at line 231 of file Module_Property.h.

6.131.4 Member Function Documentation

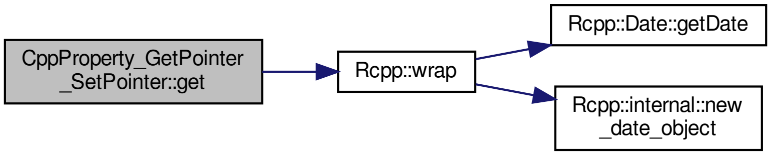
6.131.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetPointer_SetPointer< Class, PROP >::get (
            Class * object ) [inline]
```

Definition at line 234 of file Module_Property.h.

References CppProperty_GetPointer_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.131.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 241 of file Module_Property.h.

References CppProperty_GetPointer_SetPointer< Class, PROP >::class_name.

6.131.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointer_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 240 of file Module_Property.h.

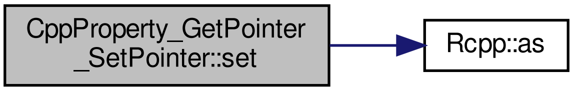
6.131.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointer_SetPointer< Class, PROP >::set (
            Class * object,
            SEXP value ) [inline]
```

Definition at line 237 of file Module_Property.h.

References Rcpp::as(), and CppProperty_GetPointer_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:



6.131.5 Member Data Documentation

6.131.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetPointer< Class, PROP >::class_name [private]
```

Definition at line 246 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetPointer< Class, PROP >::get_class().

6.131.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::getter [private]
```

Definition at line 244 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetPointer< Class, PROP >::get().

6.131.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::setter  [private]
```

Definition at line 245 of file Module_Property.h.

Referenced by CppProperty_GetPointer_SetPointer< Class, PROP >::set().

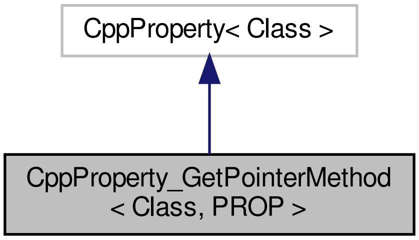
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

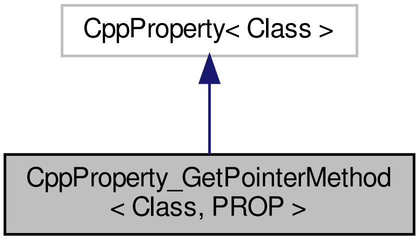
6.132 CppProperty_GetPointerMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty_GetPointerMethod< Class, PROP >:



Collaboration diagram for CppProperty_GetPointerMethod< Class, PROP >:



Public Types

- typedef PROP(∗ GetMethod) (Class ∗)
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_GetPointerMethod (GetMethod getter_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗, SEXP)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- GetMethod getter
- std::string class_name

6.132.1 Detailed Description

template<typename Class, typename PROP>
class CppProperty_GetPointerMethod< Class, PROP >

Definition at line 70 of file Module_Property.h.

6.132.2 Member Typedef Documentation

6.132.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(∗ CppProperty_GetPointerMethod< Class, PROP >::GetMethod) (Class ∗)
```

Definition at line 72 of file Module_Property.h.

6.132.2.2 prop_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetPointerMethod< Class, PROP >::prop_class
```

Definition at line 73 of file Module_Property.h.

6.132.3 Constructor & Destructor Documentation

6.132.3.1 CppProperty_GetPointerMethod()

```
template<typename Class , typename PROP >
CppProperty_GetPointerMethod< Class, PROP >::CppProperty_GetPointerMethod (
             GetMethod getter_,
             const char * doc = 0 )  [inline]
```

Definition at line 75 of file Module_Property.h.

6.132.4 Member Function Documentation

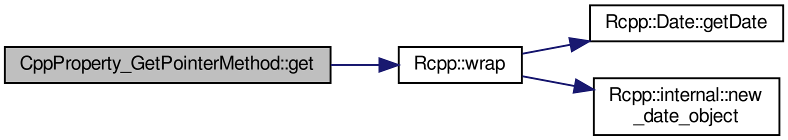
6.132.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetPointerMethod< Class, PROP >::get (
             Class * object )  [inline]
```

Definition at line 78 of file Module_Property.h.

References CppProperty_GetPointerMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



6.132.4.2 get_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointerMethod< Class, PROP >::get_class ( )  [inline]
```

Definition at line 81 of file Module_Property.h.

References CppProperty_GetPointerMethod< Class, PROP >::class_name.

6.132.4.3 is_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointerMethod< Class, PROP >::is_readonly ( )  [inline]
```

Definition at line 80 of file Module_Property.h.

6.132.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointerMethod< Class, PROP >::set (
            Class * ,
            SEXP  )  [inline]
```

Definition at line 79 of file Module_Property.h.

6.132.5 Member Data Documentation

6.132.5.1 class_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointerMethod< Class, PROP >::class_name  [private]
```

Definition at line 85 of file Module_Property.h.

Referenced by CppProperty_GetPointerMethod< Class, PROP >::get_class().

6.132.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointerMethod< Class, PROP >::getter  [private]
```

Definition at line 84 of file Module_Property.h.

Referenced by CppProperty_GetPointerMethod< Class, PROP >::get().

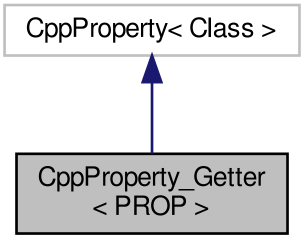
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Property.h

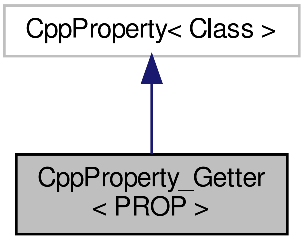
6.133 CppProperty_Getter< PROP > Class Template Reference

```
#include <Module_Field.h>
```

Inheritance diagram for CppProperty_Getter< PROP >:



Collaboration diagram for CppProperty_Getter< PROP >:



Public Types

- typedef PROP Class::∗ pointer
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_Getter (pointer ptr_, const char ∗doc=0)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- pointer ptr
- std::string class_name

6.133.1 Detailed Description

template<typename PROP>
class CppProperty_Getter< PROP >

Definition at line 48 of file Module_Field.h.

6.133.2 Member Typedef Documentation

6.133.2.1 pointer

```
template<typename PROP >
typedef PROP Class::∗ CppProperty_Getter< PROP >::pointer
```

Definition at line 50 of file Module_Field.h.

6.133.2.2 prop_class

```
template<typename PROP >
typedef CppProperty<Class> CppProperty_Getter< PROP >::prop_class
```

Definition at line 51 of file Module_Field.h.

6.133.3 Constructor & Destructor Documentation

6.133.3.1 CppProperty_Getter()

```
template<typename PROP >
CppProperty_Getter< PROP >::CppProperty_Getter (
             pointer ptr_,
             const char * doc = 0 )  [inline]
```

Definition at line 53 of file Module_Field.h.

6.133.4 Member Function Documentation

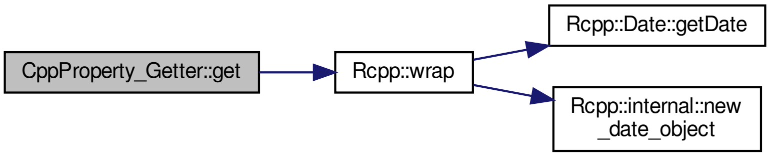
6.133.4.1 get()

```
template<typename PROP >
SEXP CppProperty_Getter< PROP >::get (
             Class * object )  [inline]
```

Definition at line 56 of file Module_Field.h.

References CppProperty_Getter< PROP >::ptr, and Rcpp::wrap().

Here is the call graph for this function:



6.133.4.2 get_class()

```
template<typename PROP >
std::string CppProperty_Getter< PROP >::get_class ( )  [inline]
```

Definition at line 59 of file Module_Field.h.

References CppProperty_Getter< PROP >::class_name.

6.133.4.3 is_readonly()

```
template<typename PROP >
bool CppProperty_Getter< PROP >::is_readonly ( )  [inline]
```

Definition at line 58 of file Module_Field.h.

6.133.4.4 set()

```
template<typename PROP >
void CppProperty_Getter< PROP >::set (
             Class * object,
             SEXP value )  [inline]
```

Definition at line 57 of file Module_Field.h.

6.133.5 Member Data Documentation

6.133.5.1 class_name

```
template<typename PROP >
std::string CppProperty_Getter< PROP >::class_name  [private]
```

Definition at line 63 of file Module_Field.h.

Referenced by CppProperty_Getter< PROP >::get_class().

6.133.5.2 ptr

```
template<typename PROP >
pointer CppProperty_Getter< PROP >::ptr  [private]
```

Definition at line 62 of file Module_Field.h.

Referenced by CppProperty_Getter< PROP >::get().

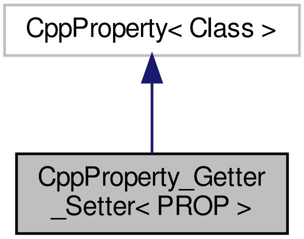
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Field.h

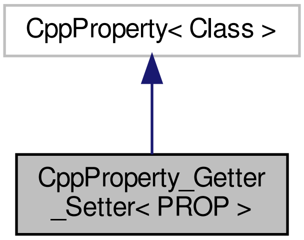
6.134 CppProperty_Getter_Setter< PROP > Class Template Reference

```
#include <Module_Field.h>
```

Inheritance diagram for CppProperty_Getter_Setter< PROP >:



Collaboration diagram for CppProperty_Getter_Setter< PROP >:



Public Types

- typedef PROP Class::∗ pointer
- typedef CppProperty< Class > prop_class

Public Member Functions

- CppProperty_Getter_Setter (pointer ptr_, const char ∗doc)
- SEXP get (Class ∗object)
- void set (Class ∗object, SEXP value)
- bool is_readonly ()
- std::string get_class ()

Private Attributes

- pointer ptr
- std::string class_name

6.134.1 Detailed Description

template<typename PROP>
class CppProperty_Getter_Setter< PROP >

Definition at line 27 of file Module_Field.h.

6.134.2 Member Typedef Documentation

6.134.2.1 pointer

```
template<typename PROP >
typedef PROP Class::∗ CppProperty_Getter_Setter< PROP >::pointer
```

Definition at line 29 of file Module_Field.h.

6.134.2.2 prop_class

```
template<typename PROP >
typedef CppProperty<Class> CppProperty_Getter_Setter< PROP >::prop_class
```

Definition at line 30 of file Module_Field.h.

6.134.3 Constructor & Destructor Documentation

6.134.3.1 CppProperty_Getter_Setter()

```
template<typename PROP >
CppProperty_Getter_Setter< PROP >::CppProperty_Getter_Setter (
             pointer ptr_,
             const char * doc )  [inline]
```

Definition at line 32 of file Module_Field.h.

6.134.4 Member Function Documentation

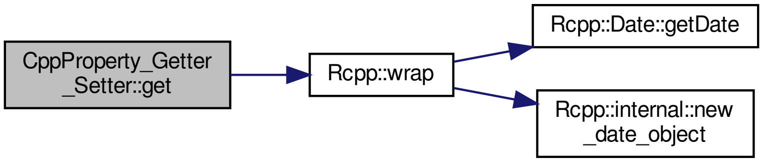
6.134.4.1 get()

```
template<typename PROP >
SEXP CppProperty_Getter_Setter< PROP >::get (
             Class * object )  [inline]
```

Definition at line 35 of file Module_Field.h.

References CppProperty_Getter_Setter< PROP >::ptr, and Rcpp::wrap().

Here is the call graph for this function:



6.134.4.2 get_class()

```
template<typename PROP >
std::string CppProperty_Getter_Setter< PROP >::get_class ( )  [inline]
```

Definition at line 38 of file Module_Field.h.

References CppProperty_Getter_Setter< PROP >::class_name.

6.134.4.3 is_readonly()

```
template<typename PROP >
bool CppProperty_Getter_Setter< PROP >::is_readonly ( )  [inline]
```

Definition at line 37 of file Module_Field.h.

6.134.4.4 set()

```
template<typename PROP >
void CppProperty_Getter_Setter< PROP >::set (
             Class * object,
             SEXP value )  [inline]
```

Definition at line 36 of file Module_Field.h.

References CppProperty_Getter_Setter< PROP >::ptr.

6.134.5 Member Data Documentation

6.134.5.1 class_name

```
template<typename PROP >
std::string CppProperty_Getter_Setter< PROP >::class_name  [private]
```

Definition at line 42 of file Module_Field.h.

Referenced by CppProperty_Getter_Setter< PROP >::get_class().

6.134.5.2 ptr

```
template<typename PROP >
pointer CppProperty_Getter_Setter< PROP >::ptr  [private]
```

Definition at line 41 of file Module_Field.h.

Referenced by CppProperty_Getter_Setter< PROP >::get(), and CppProperty_Getter_Setter< PROP >::set().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module_Field.h

6.135 Rcpp::algorithm::helpers::ctype< T > Struct Template Reference

```
#include <algorithm.h>
```

Public Types

- typedef ctype_helper< sizeof(test(make())) >::type type

Static Public Member Functions

- static CTYPE_CHAR test (const char &)
- static CTYPE_SHORT test (const short &)
- static CTYPE_INT test (const int &)
- static CTYPE_LONG test (const long &)
- static CTYPE_FLOAT test (const float &)
- static CTYPE_DOUBLE test (const double &)
- static CTYPE_LONG_DOUBLE test (const long double &)
- static CTYPE_STRING test (const std::string &)
- static CTYPE_UNSIGNED_CHAR test (const unsigned char &)
- static CTYPE_UNSIGNED_SHORT test (const unsigned short &)
- static CTYPE_UNSIGNED_INT test (const unsigned int &)
- static CTYPE_UNSIGNED_LONG test (const unsigned long &)
- static CTYPE_UNKNOWN test (...)
- static T make ()

6.135.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::ctype< T >

Definition at line 108 of file algorithm.h.

6.135.2 Member Typedef Documentation

6.135.2.1 type

```
template<typename T >
typedef ctype_helper< sizeof(test(make())) >::type Rcpp::algorithm::helpers::ctype< T >::type
```

Definition at line 132 of file algorithm.h.

6.135.3 Member Function Documentation

6.135.3.1 make()

```
template<typename T >
static T Rcpp::algorithm::helpers::ctype< T >::make ( )  [static]
```

6.135.3.2 test() [1/13]

```
template<typename T >
static CTYPE_UNKNOWN Rcpp::algorithm::helpers::ctype< T >::test (
              ... )  [static]
```

6.135.3.3 test() [2/13]

```
template<typename T >
static CTYPE_CHAR Rcpp::algorithm::helpers::ctype< T >::test (
             const char & )  [static]
```

6.135.3.4 test() [3/13]

```
template<typename T >
static CTYPE_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (
             const double & )  [static]
```

6.135.3.5 test() [4/13]

```
template<typename T >
static CTYPE_FLOAT Rcpp::algorithm::helpers::ctype< T >::test (
             const float & ) [static]
```

6.135.3.6 test() [5/13]

```
template<typename T >
static CTYPE_INT Rcpp::algorithm::helpers::ctype< T >::test (
             const int & ) [static]
```

6.135.3.7 test() [6/13]

```
template<typename T >
static CTYPE_LONG Rcpp::algorithm::helpers::ctype< T >::test (
             const long & ) [static]
```

6.135.3.8 test() [7/13]

```
template<typename T >
static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (
             const long double & ) [static]
```

6.135.3.9 test() [8/13]

```
template<typename T >
static CTYPE_SHORT Rcpp::algorithm::helpers::ctype< T >::test (
             const short & ) [static]
```

6.135.3.10 test() [9/13]

```
template<typename T >
static CTYPE_STRING Rcpp::algorithm::helpers::ctype< T >::test (
             const std::string & ) [static]
```

6.135.3.11 test() [10/13]

```
template<typename T >
static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::ctype< T >::test (
             const unsigned char & )  [static]
```

6.135.3.12 test() [11/13]

```
template<typename T >
static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::ctype< T >::test (
             const unsigned int & )  [static]
```

6.135.3.13 test() [12/13]

```
template<typename T >
static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::ctype< T >::test (
             const unsigned long & )  [static]
```

6.135.3.14 test() [13/13]

```
template<typename T >
static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::ctype< T >::test (
             const unsigned short & )  [static]
```

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.136 Rcpp::algorithm::helpers::CTYPE_CHAR Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [1]

6.136.1 Detailed Description

Definition at line 37 of file algorithm.h.

6.136.2 Member Data Documentation

6.136.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_CHAR::a[1]
```

Definition at line 37 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.137 Rcpp::algorithm::helpers::CTYPE_DOUBLE Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [7]

6.137.1 Detailed Description

Definition at line 45 of file algorithm.h.

6.137.2 Member Data Documentation

6.137.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_DOUBLE::a[7]
```

Definition at line 45 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.138 Rcpp::algorithm::helpers::CTYPE_FLOAT Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [6]

6.138.1 Detailed Description

Definition at line 44 of file algorithm.h.

6.138.2 Member Data Documentation

6.138.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_FLOAT::a[6]
```

Definition at line 44 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.139 Rcpp::algorithm::helpers::ctype_helper< I > Struct Template Reference

```
#include <algorithm.h>
```

Static Public Attributes

- static const bool value = false

6.139.1 Detailed Description

template<std::size_t I>
struct Rcpp::algorithm::helpers::ctype_helper< I >

Definition at line 58 of file algorithm.h.

6.139.2 Member Data Documentation

6.139.2.1 value

```
template<std::size_t I>
const bool Rcpp::algorithm::helpers::ctype_helper< I >::value = false  [static]
```

Definition at line 58 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.140 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef char type

Static Public Attributes

- static const bool value = true

6.140.1 Detailed Description

Definition at line 61 of file algorithm.h.

6.140.2 Member Typedef Documentation

6.140.2.1 type

```
typedef char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::type
```

Definition at line 61 of file algorithm.h.

6.140.3 Member Data Documentation

6.140.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::value = true  [static]
```

Definition at line 61 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.141 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef double type

Static Public Attributes

- static const bool value = true

6.141.1 Detailed Description

Definition at line 81 of file algorithm.h.

6.141.2 Member Typedef Documentation

6.141.2.1 type

```
typedef double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::type
```

Definition at line 81 of file algorithm.h.

6.141.3 Member Data Documentation

6.141.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::value = true  [static]
```

Definition at line 81 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.142 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef float type

Static Public Attributes

- static const bool value = true

6.142.1 Detailed Description

Definition at line 78 of file algorithm.h.

6.142.2 Member Typedef Documentation

6.142.2.1 type

```
typedef float Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::type
```

Definition at line 78 of file algorithm.h.

6.142.3 Member Data Documentation

6.142.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::value = true  [static]
```

Definition at line 78 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.143 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef int type

Static Public Attributes

- static const bool value = true

6.143.1 Detailed Description

Definition at line 67 of file algorithm.h.

6.143.2 Member Typedef Documentation

6.143.2.1 type

```
typedef int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::type
```

Definition at line 67 of file algorithm.h.

6.143.3 Member Data Documentation

6.143.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::value = true  [static]
```

Definition at line 67 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.144 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef long type

Static Public Attributes

- static const bool value = true

6.144.1 Detailed Description

Definition at line 70 of file algorithm.h.

6.144.2 Member Typedef Documentation

6.144.2.1 type

```
typedef long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::type
```

Definition at line 70 of file algorithm.h.

6.144.3 Member Data Documentation

6.144.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::value = true  [static]
```

Definition at line 70 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.145 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef long double type

Static Public Attributes

- static const bool value = true

6.145.1 Detailed Description

Definition at line 84 of file algorithm.h.

6.145.2 Member Typedef Documentation

6.145.2.1 type

```
typedef long double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::type
```

Definition at line 84 of file algorithm.h.

6.145.3 Member Data Documentation

6.145.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::value = true
[static]
```

Definition at line 84 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.146 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef short type

Static Public Attributes

- static const bool value = true

6.146.1 Detailed Description

Definition at line 64 of file algorithm.h.

6.146.2 Member Typedef Documentation

6.146.2.1 type

```
typedef short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::type
```

Definition at line 64 of file algorithm.h.

6.146.3 Member Data Documentation

6.146.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::value = true  [static]
```

Definition at line 64 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.147 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef std::string type

Static Public Attributes

- static const bool value = true

6.147.1 Detailed Description

Definition at line 87 of file algorithm.h.

6.147.2 Member Typedef Documentation

6.147.2.1 type

```
typedef std::string Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::type
```

Definition at line 87 of file algorithm.h.

6.147.3 Member Data Documentation

6.147.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::value = true  [static]
```

Definition at line 87 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.148 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef unsigned char type

Static Public Attributes

- static const bool value = true

6.148.1 Detailed Description

Definition at line 90 of file algorithm.h.

6.148.2 Member Typedef Documentation

6.148.2.1 type

```
typedef unsigned char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::type
```

Definition at line 90 of file algorithm.h.

6.148.3 Member Data Documentation

6.148.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::value = true
[static]
```

Definition at line 90 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.149 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef unsigned int type

Static Public Attributes

- static const bool value = true

6.149.1 Detailed Description

Definition at line 96 of file algorithm.h.

6.149.2 Member Typedef Documentation

6.149.2.1 type

```
typedef unsigned int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::type
```

Definition at line 96 of file algorithm.h.

6.149.3 Member Data Documentation

6.149.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::value = true
[static]
```

Definition at line 96 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.150 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef unsigned long type

Static Public Attributes

- static const bool value = true

6.150.1 Detailed Description

Definition at line 99 of file algorithm.h.

6.150.2 Member Typedef Documentation

6.150.2.1 type

```
typedef unsigned long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::type
```

Definition at line 99 of file algorithm.h.

6.150.3 Member Data Documentation

6.150.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::value = true
[static]
```

Definition at line 99 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.151 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef unsigned short type

Static Public Attributes

- static const bool value = true

6.151.1 Detailed Description

Definition at line 93 of file algorithm.h.

6.151.2 Member Typedef Documentation

6.151.2.1 type

```
typedef unsigned short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >↩
::type
```

Definition at line 93 of file algorithm.h.

6.151.3 Member Data Documentation

6.151.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >::value = true
[static]
```

Definition at line 93 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.152 Rcpp::algorithm::helpers::CTYPE_INT Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [3]

6.152.1 Detailed Description

Definition at line 39 of file algorithm.h.

6.152.2 Member Data Documentation

6.152.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_INT::a[3]
```

Definition at line 39 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.153 Rcpp::algorithm::helpers::CTYPE_LONG Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [4]

6.153.1 Detailed Description

Definition at line 40 of file algorithm.h.

6.153.2 Member Data Documentation

6.153.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_LONG::a[4]
```

Definition at line 40 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.154 Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [8]

6.154.1 Detailed Description

Definition at line 46 of file algorithm.h.

6.154.2 Member Data Documentation

6.154.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE::a[8]
```

Definition at line 46 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.155 Rcpp::algorithm::helpers::CTYPE_SHORT Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [2]

6.155.1 Detailed Description

Definition at line 38 of file algorithm.h.

6.155.2 Member Data Documentation

6.155.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_SHORT::a[2]
```

Definition at line 38 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.156 Rcpp::algorithm::helpers::CTYPE_STRING Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [9]

6.156.1 Detailed Description

Definition at line 47 of file algorithm.h.

6.156.2 Member Data Documentation

6.156.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_STRING::a[9]
```

Definition at line 47 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.157 Rcpp::algorithm::helpers::CTYPE_UNKNOWN Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [128]

6.157.1 Detailed Description

Definition at line 55 of file algorithm.h.

6.157.2 Member Data Documentation

6.157.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNKNOWN::a[128]
```

Definition at line 55 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.158 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [10]

6.158.1 Detailed Description

Definition at line 48 of file algorithm.h.

6.158.2 Member Data Documentation

6.158.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR::a[10]
```

Definition at line 48 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.159 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [12]

6.159.1 Detailed Description

Definition at line 50 of file algorithm.h.

6.159.2 Member Data Documentation

6.159.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT::a[12]
```

Definition at line 50 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.160 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [13]

6.160.1 Detailed Description

Definition at line 51 of file algorithm.h.

6.160.2 Member Data Documentation

6.160.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG::a[13]
```

Definition at line 51 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.161 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT Struct Reference

```
#include <algorithm.h>
```

Public Attributes

- char a [11]

6.161.1 Detailed Description

Definition at line 49 of file algorithm.h.

6.161.2 Member Data Documentation

6.161.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT::a[11]
```

Definition at line 49 of file algorithm.h.

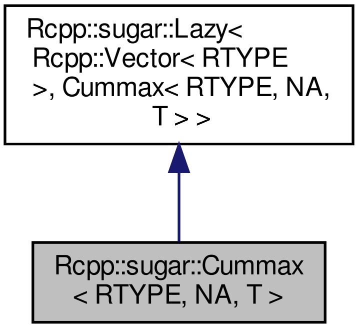
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

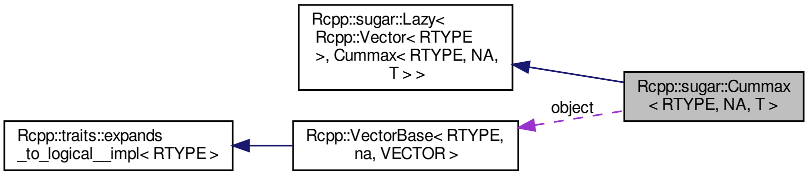
6.162 Rcpp::sugar::Cummax< RTYPE, NA, T > Class Template Reference

```
#include <cummax.h>
```

Inheritance diagram for Rcpp::sugar::Cummax< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummax< RTYPE, NA, T > >
Rcpp::sugar::Cummax < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Cummax< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummax< RTYPE, NA, T > >
Rcpp::sugar::Cummax < RTYPE, NA, T >
object_
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::Vector< RTYPE > VECTOR

Public Member Functions

- Cummax (const VEC_TYPE &object_)
- VECTOR get () const

Private Attributes

- const VEC_TYPE & object

6.162.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cummax< RTYPE, NA, T >

Definition at line 29 of file cummax.h.

6.162.2 Member Typedef Documentation

6.162.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cummax< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cummax.h.

6.162.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cummax< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cummax.h.

6.162.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cummax< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cummax.h.

6.162.3 Constructor & Destructor Documentation

6.162.3.1 Cummax()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cummax< RTYPE, NA, T >::Cummax (
            const VEC_TYPE & object_ )  [inline]
```

Definition at line 35 of file cummax.h.

6.162.4 Member Function Documentation

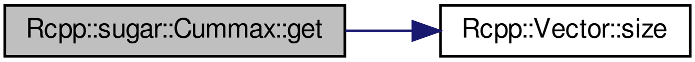
6.162.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cummax< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 37 of file cummax.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.162.5 Member Data Documentation

6.162.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cummax< RTYPE, NA, T >::object  [private]
```

Definition at line 52 of file cummax.h.

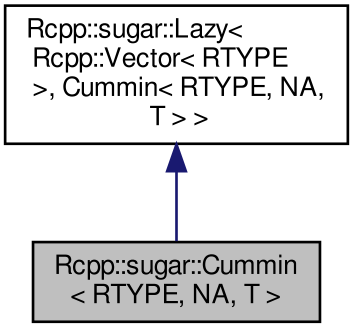
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cummax.h

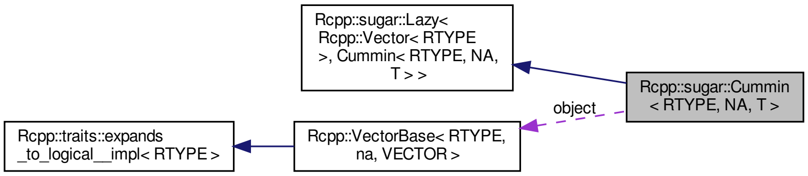
6.163 Rcpp::sugar::Cummin< RTYPE, NA, T > Class Template Reference

```
#include <cummin.h>
```

Inheritance diagram for Rcpp::sugar::Cummin< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummin< RTYPE, NA, T > >
Rcpp::sugar::Cummin < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Cummin< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummin< RTYPE, NA, T > >
Rcpp::sugar::Cummin < RTYPE, NA, T >
object_
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::Vector< RTYPE > VECTOR

Public Member Functions

- Cummin (const VEC_TYPE &object_)
- VECTOR get () const

Private Attributes

- const VEC_TYPE & object

6.163.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cummin< RTYPE, NA, T >

Definition at line 29 of file cummin.h.

6.163.2 Member Typedef Documentation

6.163.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cummin< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cummin.h.

6.163.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cummin< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cummin.h.

6.163.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cummin< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cummin.h.

6.163.3 Constructor & Destructor Documentation

6.163.3.1 Cummin()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cummin< RTYPE, NA, T >::Cummin (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 35 of file cummin.h.

6.163.4 Member Function Documentation

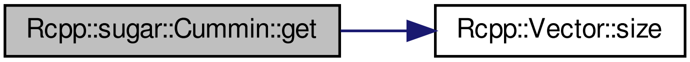
6.163.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cummin< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 37 of file cummin.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.163.5 Member Data Documentation

6.163.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cummin< RTYPE, NA, T >::object  [private]
```

Definition at line 52 of file cummin.h.

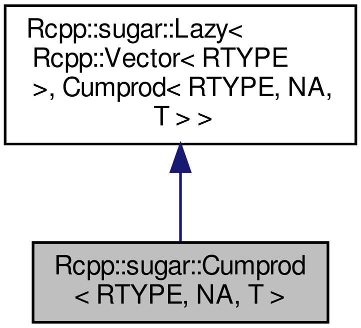
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cummin.h

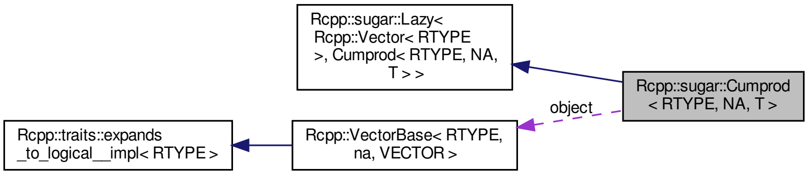
6.164 Rcpp::sugar::Cumprod< RTYPE, NA, T > Class Template Reference

```
#include <cumprod.h>
```

Inheritance diagram for Rcpp::sugar::Cumprod< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumprod< RTYPE, NA, T > >
Rcpp::sugar::Cumprod < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Cumprod< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumprod< RTYPE, NA, T > >
Rcpp::sugar::Cumprod < RTYPE, NA, T >
object_
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::Vector< RTYPE > VECTOR

Public Member Functions

- Cumprod (const VEC_TYPE &object_)
- VECTOR get () const

Private Attributes

- const VEC_TYPE & object

6.164.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cumprod< RTYPE, NA, T >

Definition at line 29 of file cumprod.h.

6.164.2 Member Typedef Documentation

6.164.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cumprod< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cumprod.h.

6.164.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cumprod< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cumprod.h.

6.164.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cumprod< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cumprod.h.

6.164.3 Constructor & Destructor Documentation

6.164.3.1 Cumprod()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cumprod< RTYPE, NA, T >::Cumprod (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 35 of file cumprod.h.

6.164.4 Member Function Documentation

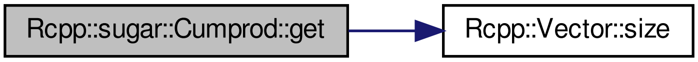
6.164.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cumprod< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 37 of file cumprod.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.164.5 Member Data Documentation

6.164.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cumprod< RTYPE, NA, T >::object  [private]
```

Definition at line 52 of file cumprod.h.

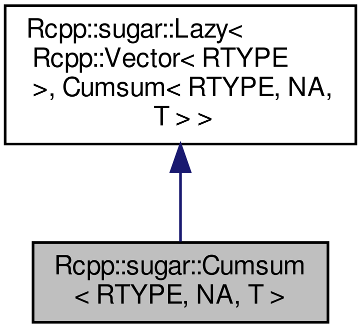
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cumprod.h

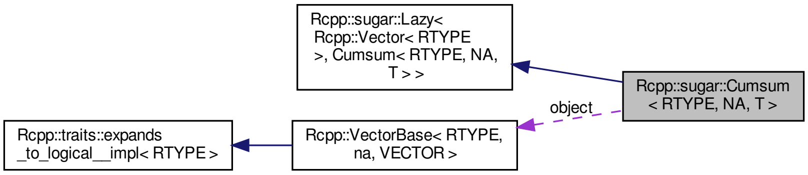
6.165 Rcpp::sugar::Cumsum< RTYPE, NA, T > Class Template Reference

```
#include <cumsum.h>
```

Inheritance diagram for Rcpp::sugar::Cumsum< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumsum< RTYPE, NA, T > >
Rcpp::sugar::Cumsum < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Cumsum< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumsum< RTYPE, NA, T > >
Rcpp::sugar::Cumsum < RTYPE, NA, T >
object_
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::Vector< RTYPE > VECTOR

Public Member Functions

- Cumsum (const VEC_TYPE &object_)
- VECTOR get () const

Private Attributes

- const VEC_TYPE & object

6.165.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Cumsum< RTYPE, NA, T >

Definition at line 29 of file cumsum.h.

6.165.2 Member Typedef Documentation

6.165.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cumsum< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cumsum.h.

6.165.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Cumsum< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cumsum.h.

6.165.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cumsum< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cumsum.h.

6.165.3 Constructor & Destructor Documentation

6.165.3.1 Cumsum()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cumsum< RTYPE, NA, T >::Cumsum (
            const VEC_TYPE & object_ )  [inline]
```

Definition at line 35 of file cumsum.h.

6.165.4 Member Function Documentation

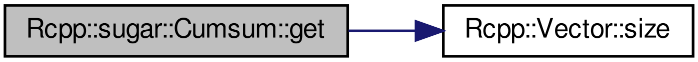
6.165.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cumsum< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 37 of file cumsum.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.165.5 Member Data Documentation

6.165.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cumsum< RTYPE, NA, T >::object  [private]
```

Definition at line 53 of file cumsum.h.

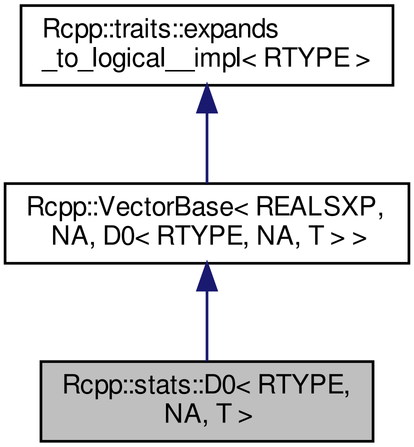
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cumsum.h

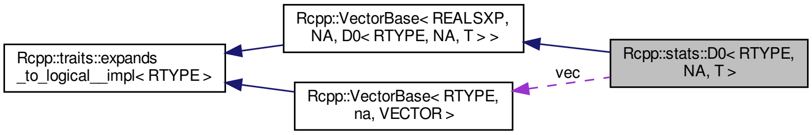
6.166 Rcpp::stats::D0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D0< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D0< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, int)

Public Member Functions

- D0 (FunPtr ptr_, const VEC_TYPE &vec_, bool log_)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- int log

6.166.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D0< RTYPE, NA, T >

Definition at line 33 of file dpq.h.

6.166.2 Member Typedef Documentation

6.166.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::D0< RTYPE, NA, T >::FunPtr) (double, int)
```

Definition at line 36 of file dpq.h.

6.166.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D0< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 35 of file dpq.h.

6.166.3 Constructor & Destructor Documentation

6.166.3.1 D0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D0< RTYPE, NA, T >::D0 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             bool log_ )  [inline]
```

Definition at line 38 of file dpq.h.

6.166.4 Member Function Documentation

6.166.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D0< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 41 of file dpq.h.

References Rcpp::stats::D0< RTYPE, NA, T >::log, Rcpp::stats::D0< RTYPE, NA, T >::ptr, and Rcpp::stats::D0< RTYPE, NA, T >::vec.

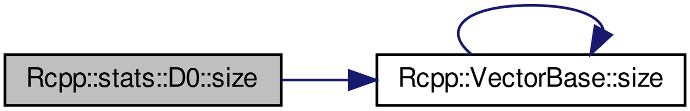
6.166.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D0< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 45 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::D0< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.166.5 Member Data Documentation

6.166.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D0< RTYPE, NA, T >::log  [private]
```

Definition at line 50 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[ ]().

6.166.5.2 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::D0< RTYPE, NA, T >::ptr  [private]
```

Definition at line 48 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[ ]().

6.166.5.3 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::D0< RTYPE, NA, T >::vec  [private]
```

Definition at line 49 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::D0< RTYPE, NA, T >::size().

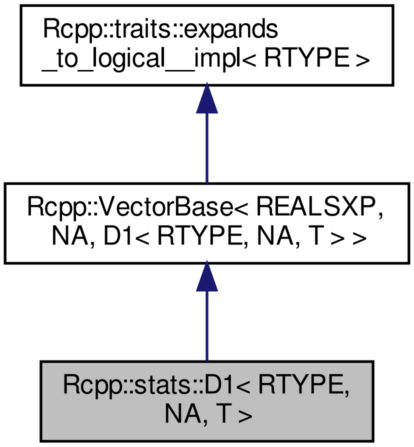
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

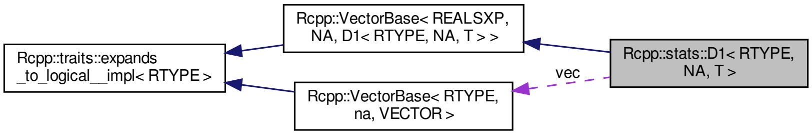
6.167 Rcpp::stats::D1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D1< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, int)

Public Member Functions

- D1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool log_)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- int log

6.167.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D1< RTYPE, NA, T >

Definition at line 54 of file dpq.h.

6.167.2 Member Typedef Documentation

6.167.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::D1< RTYPE, NA, T >::FunPtr) (double, double, int)
```

Definition at line 57 of file dpq.h.

6.167.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D1< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 56 of file dpq.h.

6.167.3 Constructor & Destructor Documentation

6.167.3.1 D1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D1< RTYPE, NA, T >::D1 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             double p0_,
             bool log_ )  [inline]
```

Definition at line 59 of file dpq.h.

6.167.4 Member Function Documentation

6.167.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D1< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 62 of file dpq.h.

References Rcpp::stats::D1< RTYPE, NA, T >::log, Rcpp::stats::D1< RTYPE, NA, T >::p0, Rcpp::stats::D1< RTYPE, NA, T >::ptr, and Rcpp::stats::D1< RTYPE, NA, T >::vec.

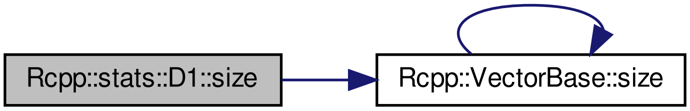
6.167.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D1< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 66 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::D1< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.167.5 Member Data Documentation

6.167.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D1< RTYPE, NA, T >::log  [private]
```

Definition at line 72 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[ ]().

6.167.5.2 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D1< RTYPE, NA, T >::p0  [private]
```

Definition at line 71 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[ ]().

6.167.5.3 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::D1< RTYPE, NA, T >::ptr  [private]
```

Definition at line 69 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[ ]().

6.167.5.4 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::D1< RTYPE, NA, T >::vec  [private]
```

Definition at line 70 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::D1< RTYPE, NA, T >::size().

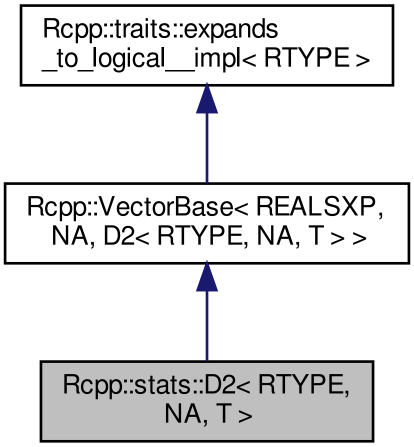
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

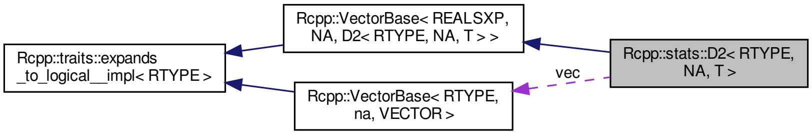
6.168 Rcpp::stats::D2< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D2< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > >
Rcpp::stats::D2< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::D2< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::D2< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, double, int)

Public Member Functions

- D2 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, double p1_, bool log_)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- double p1
- int log

6.168.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D2< RTYPE, NA, T >

Definition at line 76 of file dpq.h.

6.168.2 Member Typedef Documentation

6.168.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::D2< RTYPE, NA, T >::FunPtr) (double, double, double, int)
```

Definition at line 79 of file dpq.h.

6.168.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D2< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 78 of file dpq.h.

6.168.3 Constructor & Destructor Documentation

6.168.3.1 D2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D2< RTYPE, NA, T >::D2 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             double p0_,
             double p1_,
             bool log_ )  [inline]
```

Definition at line 81 of file dpq.h.

6.168.4 Member Function Documentation

6.168.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D2< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 84 of file dpq.h.

References Rcpp::stats::D2< RTYPE, NA, T >::log, Rcpp::stats::D2< RTYPE, NA, T >::p0, Rcpp::stats::D2< RTYPE, NA, T >::p1, Rcpp::stats::D2< RTYPE, NA, T >::ptr, and Rcpp::stats::D2< RTYPE, NA, T >::vec.

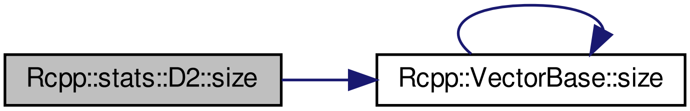
6.168.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D2< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 88 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::D2< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.168.5 Member Data Documentation

6.168.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D2< RTYPE, NA, T >::log  [private]
```

Definition at line 94 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[ ]().

6.168.5.2 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D2< RTYPE, NA, T >::p0  [private]
```

Definition at line 93 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[ ]().

6.168.5.3 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D2< RTYPE, NA, T >::p1  [private]
```

Definition at line 93 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[ ]().

6.168.5.4 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::D2< RTYPE, NA, T >::ptr  [private]
```

Definition at line 91 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[ ]().

6.168.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::D2< RTYPE, NA, T >::vec  [private]
```

Definition at line 92 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::D2< RTYPE, NA, T >::size().

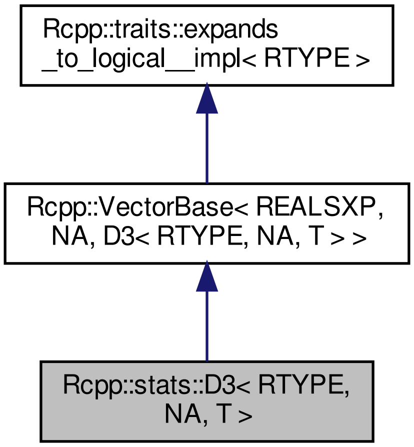
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

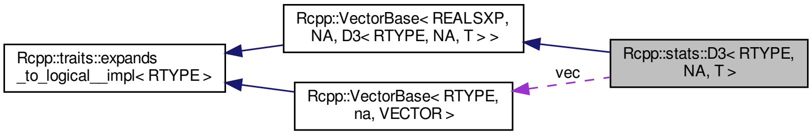
6.169 Rcpp::stats::D3< RTYPE, NA, T > Class Template Reference

```
#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

6.169.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D3< RTYPE, NA, T >

Definition at line 98 of file dpq.h.

6.169.2 Member Typedef Documentation

6.169.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(∗ Rcpp::stats::D3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int)
```

Definition at line 101 of file dpq.h.

6.169.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D3< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 100 of file dpq.h.

6.169.3 Constructor & Destructor Documentation

6.169.3.1 D3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D3< RTYPE, NA, T >::D3 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             double p0_,
             double p1_,
             double p2_,
             bool log_ )  [inline]
```

Definition at line 103 of file dpq.h.

6.169.4 Member Function Documentation

6.169.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 106 of file dpq.h.

References Rcpp::stats::D3< RTYPE, NA, T >::log, Rcpp::stats::D3< RTYPE, NA, T >::p0, Rcpp::stats::D3< RTYPE, NA, T >::p1, Rcpp::stats::D3< RTYPE, NA, T >::p2, Rcpp::stats::D3< RTYPE, NA, T >::ptr, and Rcpp::stats::D3< RTYPE, NA, T >::vec.

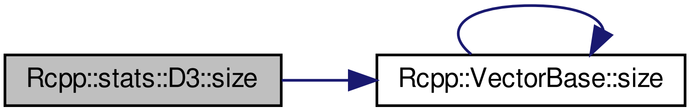
6.169.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D3< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 110 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::D3< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.169.5 Member Data Documentation

6.169.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D3< RTYPE, NA, T >::log  [private]
```

Definition at line 116 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ]().

6.169.5.2 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::p0  [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ]().

6.169.5.3 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::p1  [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ]().

6.169.5.4 p2

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::p2  [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ]().

6.169.5.5 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::D3< RTYPE, NA, T >::ptr  [private]
```

Definition at line 113 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ]().

6.169.5.6 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::D3< RTYPE, NA, T >::vec  [private]
```

Definition at line 114 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::D3< RTYPE, NA, T >::size().

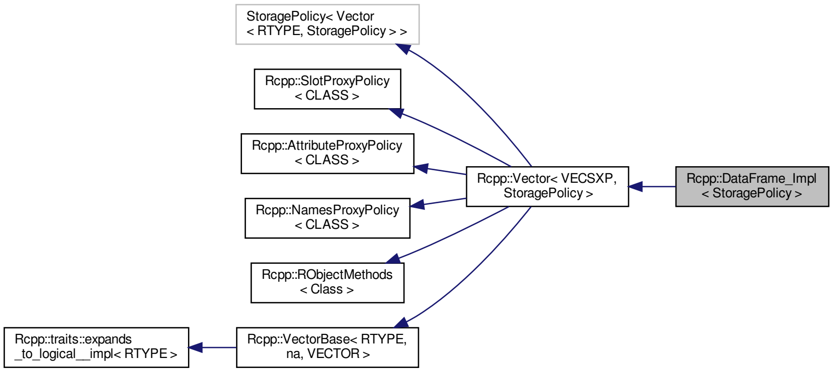
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

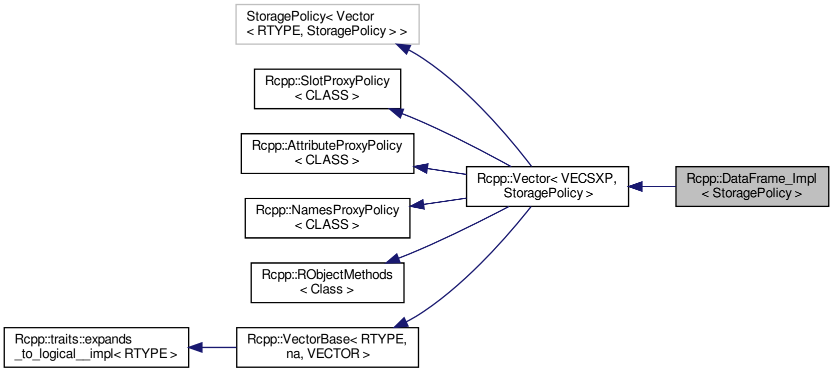
6.170 Rcpp::DataFrame_Impl< StoragePolicy > Class Template Reference

```
#include <DataFrame.h>
```

Inheritance diagram for Rcpp::DataFrame_Impl< StoragePolicy >:


StoragePolicy< Vector < RTYPE, StoragePolicy > >
Rcpp::SlotProxyPolicy < CLASS >
Rcpp::AttributeProxyPolicy < CLASS >
Rcpp::NamesProxyPolicy < CLASS >
Rcpp::RObjectMethods < Class >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >
Rcpp::Vector< VECSXP, StoragePolicy >
Rcpp::DataFrame_Impl < StoragePolicy >

Collaboration diagram for Rcpp::DataFrame_Impl< StoragePolicy >:


StoragePolicy< Vector < RTYPE, StoragePolicy > >
Rcpp::SlotProxyPolicy < CLASS >
Rcpp::AttributeProxyPolicy < CLASS >
Rcpp::NamesProxyPolicy < CLASS >
Rcpp::RObjectMethods < Class >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >
Rcpp::Vector< VECSXP, StoragePolicy >
Rcpp::DataFrame_Impl < StoragePolicy >

Public Types

- typedef Vector< VECSXP, StoragePolicy > Parent

Public Member Functions

- DataFrame_Impl ()
- DataFrame_Impl (SEXP x)

- DataFrame_Impl (const DataFrame_Impl &other)
- template<typename T >
  DataFrame_Impl (const T &obj)
- DataFrame_Impl & operator= (DataFrame_Impl &other)
- DataFrame_Impl & operator= (SEXP x)
- int nrow () const
- template<typename T >
  void push_back (const T &object)
- template<typename T >
  void push_back (const T &object, const std::string &name)
- template<typename T >
  void push_front (const T &object)
- template<typename T >
  void push_front (const T &object, const std::string &name)
- int nrows () const
- int rows () const
- R_xlen_t ncol () const
- R_xlen_t cols () const

Static Public Member Functions

- static DataFrame_Impl create ()

Private Member Functions

- void set__ (SEXP x)
- void set_type_after_push ()

Static Private Member Functions

- static DataFrame_Impl from_list (Parent obj)

Additional Inherited Members

6.170.1 Detailed Description

template<template< class > class StoragePolicy>
class Rcpp::DataFrame_Impl< StoragePolicy >

Definition at line 38 of file DataFrame.h.

6.170.2 Member Typedef Documentation

6.170.2.1 Parent

```
template<template< class > class StoragePolicy>
typedef Vector<VECSXP, StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::Parent
```

Definition at line 40 of file DataFrame.h.

6.170.3 Constructor & Destructor Documentation

6.170.3.1 DataFrame_Impl() [1/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl ( )  [inline]
```

Definition at line 42 of file DataFrame.h.

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::create(), and Rcpp::DataFrame_Impl< StoragePolicy >::from_list().

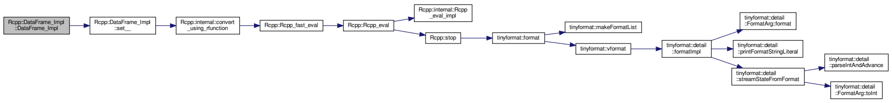
6.170.3.2 DataFrame_Impl() [2/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
             SEXP x )  [inline]
```

Definition at line 43 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::set__().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::DataFrame_Impl
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

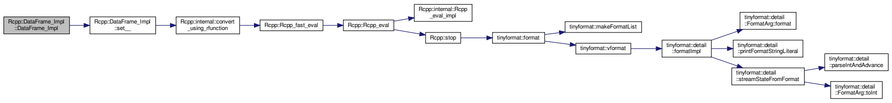
6.170.3.3 DataFrame_Impl() [3/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
             const DataFrame_Impl< StoragePolicy > & other )  [inline]
```

Definition at line 46 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::set__().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::DataFrame_Impl
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

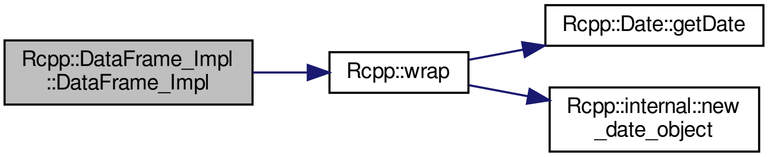
6.170.3.4 DataFrame_Impl() [4/4]

```
template<template< class > class StoragePolicy>
template<class T >
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
             const T & obj )
```

Definition at line 25 of file DataFrame.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::DataFrame_Impl
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.170.4 Member Function Documentation

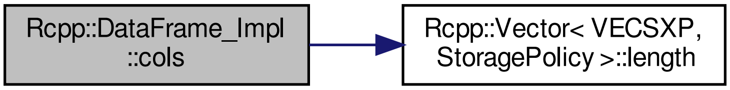
6.170.4.1 cols()

```
template<template< class > class StoragePolicy>
R_xlen_t Rcpp::DataFrame_Impl< StoragePolicy >::cols ( ) const  [inline]
```

Definition at line 115 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::length().

Here is the call graph for this function:



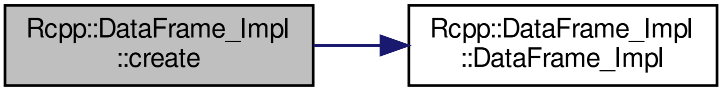
6.170.4.2 create()

```
template<template< class > class StoragePolicy>
static DataFrame_Impl Rcpp::DataFrame_Impl< StoragePolicy >::create ( )  [inline], [static]
```

Definition at line 117 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl().

Here is the call graph for this function:



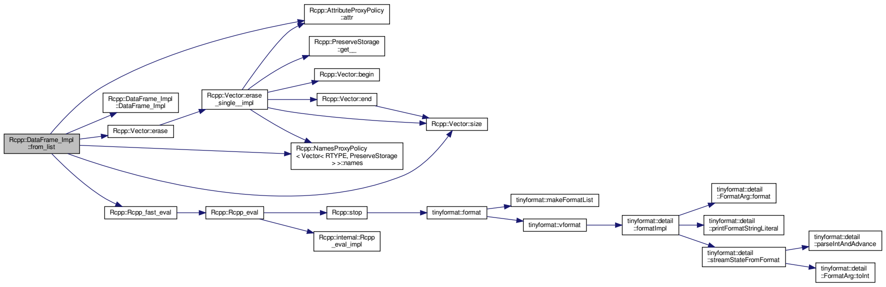
6.170.4.3 from_list()

```
template<template< class > class StoragePolicy>
static DataFrame_Impl Rcpp::DataFrame_Impl< StoragePolicy >::from_list (
             Parent obj ) [inline], [static], [private]
```

Definition at line 156 of file DataFrame.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_↩
Impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase(), Rcpp::NamesProxyPolicy< CLASS >::names(), Rcpp::↩
Rcpp_fast_eval(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



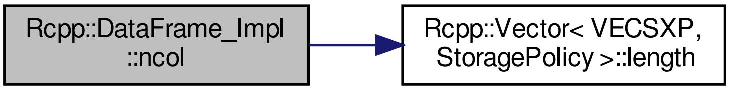
6.170.4.4 ncol()

```
template<template< class > class StoragePolicy>
R_xlen_t Rcpp::DataFrame_Impl< StoragePolicy >::ncol ( ) const [inline]
```

Definition at line 114 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::length().

Here is the call graph for this function:



6.170.4.5 nrow()

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::nrow ( ) const  [inline]
```

Definition at line 69 of file DataFrame.h.

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::nrows(), and Rcpp::DataFrame_Impl< StoragePolicy >↩ ::rows().

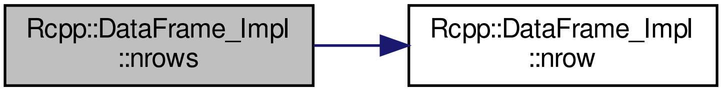
6.170.4.6 nrows()

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::nrows ( ) const  [inline]
```

Definition at line 111 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::nrow().

Here is the call graph for this function:



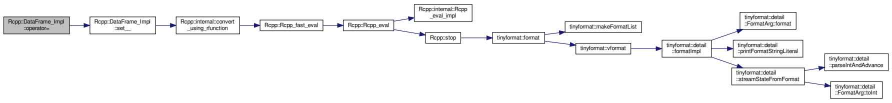
6.170.4.7 operator=() [1/2]

```
template<template< class > class StoragePolicy>
DataFrame_Impl& Rcpp::DataFrame_Impl< StoragePolicy >::operator= (
              DataFrame_Impl< StoragePolicy > & other )  [inline]
```

Definition at line 53 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::set__().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::operator=
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

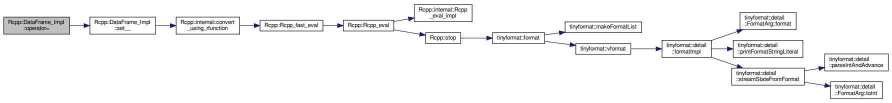
6.170.4.8 operator=() [2/2]

```
template<template< class > class StoragePolicy>
DataFrame_Impl& Rcpp::DataFrame_Impl< StoragePolicy >::operator= (
             SEXP x ) [inline]
```

Definition at line 58 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::set__().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::operator=
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

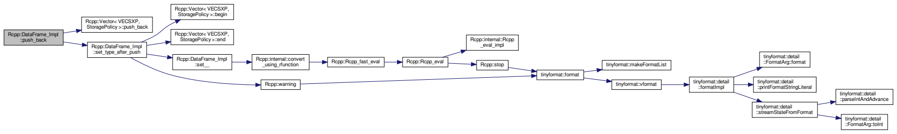
6.170.4.9 push_back() [1/2]

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_back (
             const T & object ) [inline]
```

Definition at line 87 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push_back(), and Rcpp::DataFrame_Impl< StoragePolicy >↩
::set_type_after_push().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::push_back
Rcpp::Vector< VECSXP, StoragePolicy >::push_back
Rcpp::DataFrame_Impl ::set_type_after_push
Rcpp::Vector< VECSXP, StoragePolicy >::begin
Rcpp::Vector< VECSXP, StoragePolicy >::end
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

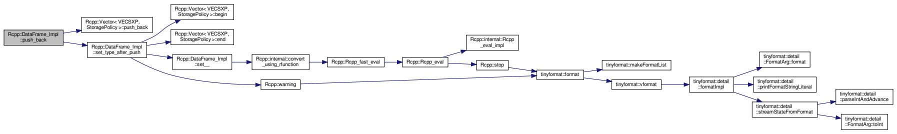
6.170.4.10 push_back() [2/2]

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_back (
            const T & object,
            const std::string & name ) [inline]
```

Definition at line 93 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push_back(), and Rcpp::DataFrame_Impl< StoragePolicy >↩
::set_type_after_push().

Here is the call graph for this function:



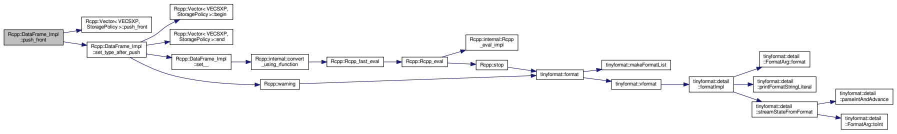
6.170.4.11 push_front() [1/2]

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_front (
            const T & object ) [inline]
```

Definition at line 99 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push_front(), and Rcpp::DataFrame_Impl< StoragePolicy >↩
::set_type_after_push().

Here is the call graph for this function:



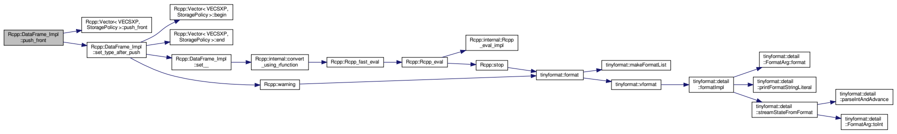
6.170.4.12 push_front() [2/2]

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_front (
             const T & object,
             const std::string & name )  [inline]
```

Definition at line 105 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push_front(), and Rcpp::DataFrame_Impl< StoragePolicy >↩ ::set_type_after_push().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::push_front
Rcpp::Vector< VECSXP, StoragePolicy >::push_front
Rcpp::DataFrame_Impl ::set_type_after_push
Rcpp::Vector< VECSXP, StoragePolicy >::begin
Rcpp::Vector< VECSXP, StoragePolicy >::end
Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

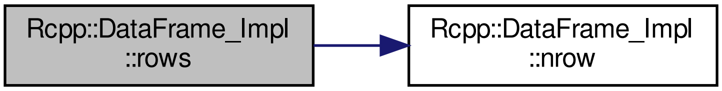
6.170.4.13 rows()

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::rows ( ) const  [inline]
```

Definition at line 112 of file DataFrame.h.

References Rcpp::DataFrame_Impl< StoragePolicy >::nrow().

Here is the call graph for this function:



6.170.4.14 set__()

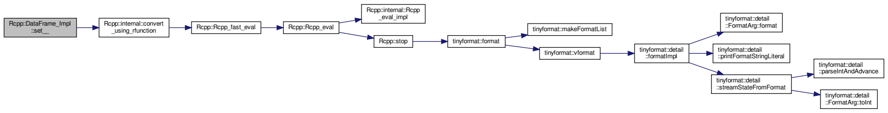
```
template<template< class > class StoragePolicy>
void Rcpp::DataFrame_Impl< StoragePolicy >::set__ (
             SEXP x )  [inline], [private]
```

Definition at line 124 of file DataFrame.h.

References Rcpp::internal::convert_using_rfunction().

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl(), Rcpp::DataFrame_Impl< StoragePolicy >::operator=(), and Rcpp::DataFrame_Impl< StoragePolicy >::set_type_after_push().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::set__
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.170.4.15 set_type_after_push()

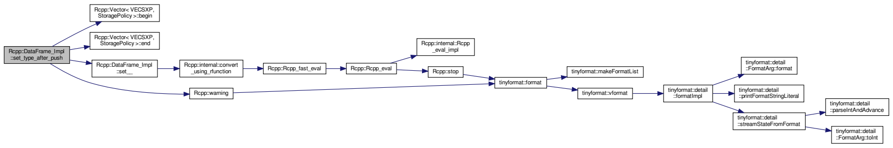
```
template<template< class > class StoragePolicy>
void Rcpp::DataFrame_Impl< StoragePolicy >::set_type_after_push ( )  [inline], [private]
```

Definition at line 133 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::begin(), Rcpp::Vector< VECSXP, StoragePolicy >::end(), Rcpp::DataFrame_Impl< StoragePolicy >::set__(), and Rcpp::warning().

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::push_back(), and Rcpp::DataFrame_Impl< StoragePolicy >::push_front().

Here is the call graph for this function:


Rcpp::DataFrame_Impl ::set_type_after_push
Rcpp::Vector< VECSXP, StoragePolicy >::begin
Rcpp::Vector< VECSXP, StoragePolicy >::end
Rcpp::DataFrame_Impl ::set__
Rcpp::warning
Rcpp::internal::convert _using_rfunction
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

The documentation for this class was generated from the following file:

- inst/include/Rcpp/DataFrame.h

6.171 Rcpp::Date Class Reference

```
#include <Date.h>
```

Public Member Functions

- Date ()
- Date (SEXP s)
- Date (const int &dt)
- Date (const double &dt)
- Date (const std::string &s, const std::string &fmt="%Y-%m-%d")
- Date (const unsigned int &mon, const unsigned int &day, const unsigned int &year)
- double getDate (void) const
- int getDay () const
- int getMonth () const
- int getYear () const
- int getWeekday () const
- int getYearday () const
- int is_na () const
- operator double () const
- std::string format (const char ∗fmt="%Y-%m-%d") const

Static Public Member Functions

- static unsigned int baseYear ()

Private Member Functions

- void update_tm ()

Private Attributes

- double m_d
- struct tm m_tm

Friends

- Date operator+ (const Date &date, int offset)
- double operator- (const Date &date1, const Date &date2)
- bool operator< (const Date &date1, const Date &date2)
- bool operator> (const Date &date1, const Date &date2)
- bool operator== (const Date &date1, const Date &date2)
- bool operator>= (const Date &date1, const Date &date2)
- bool operator<= (const Date &date1, const Date &date2)
- bool operator!= (const Date &date1, const Date &date2)
- std::ostream & operator<< (std::ostream &os, const Date d)

6.171.1 Detailed Description

Definition at line 31 of file Date.h.

6.171.2 Constructor & Destructor Documentation

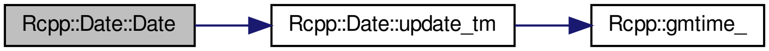
6.171.2.1 Date() [1/6]

```
Rcpp::Date::Date ( )  [inline]
```

Definition at line 33 of file Date.h.

References m_d, and update_tm().

Here is the call graph for this function:



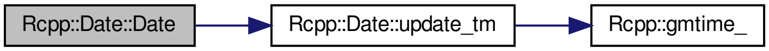
6.171.2.2 Date() [2/6]

```
Rcpp::Date::Date (
              SEXP s )  [inline]
```

Definition at line 25 of file Date.h.

References m_d, and update_tm().

Here is the call graph for this function:



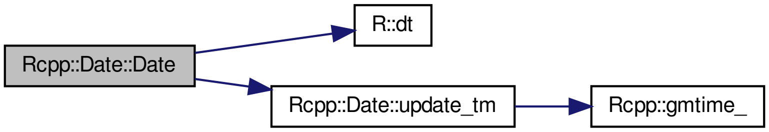
6.171.2.3 Date() [3/6]

```
Rcpp::Date::Date (
            const int & dt )  [inline]
```

Definition at line 40 of file Date.h.

References R::dt(), m_d, and update_tm().

Here is the call graph for this function:



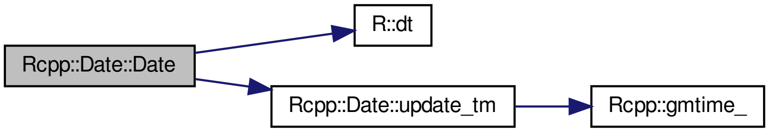
6.171.2.4 Date() [4/6]

```
Rcpp::Date::Date (
            const double & dt )  [inline]
```

Definition at line 46 of file Date.h.

References R::dt(), m_d, and update_tm().

Here is the call graph for this function:



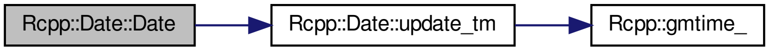
6.171.2.5 Date() [5/6]

```
Rcpp::Date::Date (
            const std::string & s,
            const std::string & fmt = "%Y-%m-%d" )  [inline]
```

Definition at line 30 of file Date.h.

References m_d, and update_tm().

Here is the call graph for this function:



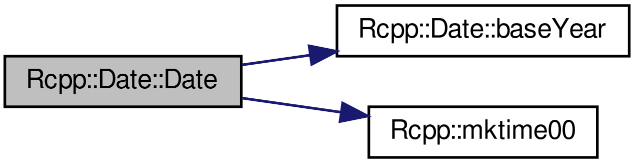
6.171.2.6 Date() [6/6]

```
Rcpp::Date::Date (
            const unsigned int & mon,
            const unsigned int & day,
            const unsigned int & year )  [inline]
```

Definition at line 52 of file Date.h.

References baseYear(), m_d, m_tm, and Rcpp::mktime00().

Here is the call graph for this function:



6.171.3 Member Function Documentation

6.171.3.1 baseYear()

```
static unsigned int Rcpp::Date::baseYear ( )  [inline], [static]
```

Definition at line 85 of file Date.h.

Referenced by Date(), and format().

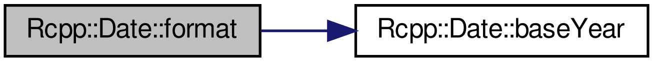
6.171.3.2 format()

```
std::string Rcpp::Date::format (
             const char * fmt = "%Y-%m-%d" ) const  [inline]
```

Definition at line 107 of file Date.h.

References baseYear(), m_tm, and Rcpp::tm.

Here is the call graph for this function:



6.171.3.3 getDate()

```
double Rcpp::Date::getDate (
             void  ) const  [inline]
```

Definition at line 70 of file Date.h.

References m_d.

Referenced by Rcpp::internal::caster< Rcpp::Date, double >(), and Rcpp::wrap().

6.171.3.4 getDay()

```
int Rcpp::Date::getDay ( ) const  [inline]
```

Definition at line 78 of file Date.h.

References m_tm.

6.171.3.5 getMonth()

```
int Rcpp::Date::getMonth ( ) const  [inline]
```

Definition at line 79 of file Date.h.

References m_tm.

6.171.3.6 getWeekday()

```
int Rcpp::Date::getWeekday ( ) const  [inline]
```

Definition at line 81 of file Date.h.

References m_tm.

6.171.3.7 getYear()

```
int Rcpp::Date::getYear ( ) const  [inline]
```

Definition at line 80 of file Date.h.

References m_tm.

6.171.3.8 getYearday()

```
int Rcpp::Date::getYearday ( ) const  [inline]
```

Definition at line 82 of file Date.h.

References m_tm.

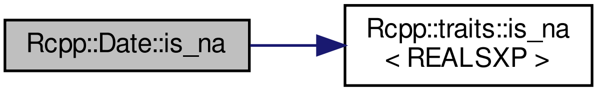
6.171.3.9 is_na()

```
int Rcpp::Date::is_na ( ) const  [inline]
```

Definition at line 99 of file Date.h.

References Rcpp::traits::is_na< REALSXP >(), and m_d.

Here is the call graph for this function:



6.171.3.10 operator double()

```
Rcpp::Date::operator double ( ) const  [inline]
```

Definition at line 103 of file Date.h.

References m_d.

6.171.3.11 update_tm()

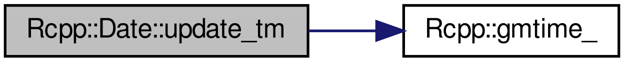
```
void Rcpp::Date::update_tm ( )  [inline], [private]
```

Definition at line 126 of file Date.h.

References Rcpp::gmtime_(), m_d, and m_tm.

Referenced by Date().

Here is the call graph for this function:



6.171.4 Friends And Related Function Documentation

6.171.4.1 operator"!=

```
bool operator!= (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 170 of file Date.h.

6.171.4.2 operator+

```
Date operator+ (
             const Date & date,
             int offset )  [friend]
```

Definition at line 156 of file Date.h.

6.171.4.3 operator-

```
double operator- (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 164 of file Date.h.

6.171.4.4 operator<

```
bool operator< (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 165 of file Date.h.

6.171.4.5 operator<<

```
std::ostream& operator<< (
             std::ostream & os,
             const Date d )  [friend]
```

Definition at line 172 of file Date.h.

6.171.4.6 operator<=

```
bool operator<= (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 169 of file Date.h.

6.171.4.7 operator==

```
bool operator== (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 167 of file Date.h.

6.171.4.8 operator>

```
bool operator> (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 166 of file Date.h.

6.171.4.9 operator>=

```
bool operator>= (
             const Date & date1,
             const Date & date2 )  [friend]
```

Definition at line 168 of file Date.h.

6.171.5 Member Data Documentation

6.171.5.1 m_d

```
double Rcpp::Date::m_d  [private]
```

Definition at line 122 of file Date.h.

Referenced by Date(), getDate(), is_na(), operator double(), and update_tm().

6.171.5.2 m_tm

```
struct tm Rcpp::Date::m_tm  [private]
```

Definition at line 122 of file Date.h.

Referenced by Date(), format(), getDay(), getMonth(), getWeekday(), getYear(), getYearday(), and update_tm().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/Date.h

6.172 Rcpp::Datetime Class Reference

```
#include <Datetime.h>
```

Public Member Functions

- Datetime ()
- Datetime (SEXP s)
- Datetime (const double &dt)
- Datetime (const std::string &s, const std::string &fmt="%Y-%m-%d %H:%M:%OS")
- double getFractionalTimestamp (void) const
- int getMicroSeconds () const
- int getSeconds () const
- int getMinutes () const
- int getHours () const
- int getDay () const
- int getMonth () const
- int getYear () const
- int getWeekday () const
- int getYearday () const
- int is_na () const
- operator double () const
- std::string format (const char ∗fmt="%Y-%m-%d %H:%M:%S") const

Private Member Functions

- void update_tm ()

Static Private Member Functions

- static unsigned int baseYear ()

Private Attributes

- double m_dt
- struct tm m_tm
- unsigned int m_us

Friends

- Datetime operator+ (const Datetime &dt, double offset)
- Datetime operator+ (const Datetime &dt, int offset)
- double operator- (const Datetime &dt1, const Datetime &dt2)
- bool operator< (const Datetime &dt1, const Datetime &dt2)
- bool operator> (const Datetime &dt1, const Datetime &dt2)
- bool operator== (const Datetime &dt1, const Datetime &dt2)
- bool operator>= (const Datetime &dt1, const Datetime &dt2)
- bool operator<= (const Datetime &dt1, const Datetime &dt2)
- bool operator!= (const Datetime &dt1, const Datetime &dt2)
- std::ostream & operator<< (std::ostream &s, const Datetime d)

6.172.1 Detailed Description

Definition at line 33 of file Datetime.h.

6.172.2 Constructor & Destructor Documentation

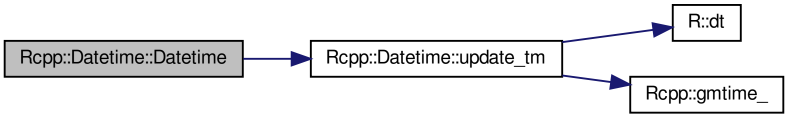
6.172.2.1 Datetime() [1/4]

```
Rcpp::Datetime::Datetime ( )  [inline]
```

Definition at line 35 of file Datetime.h.

References m_dt, and update_tm().

Here is the call graph for this function:



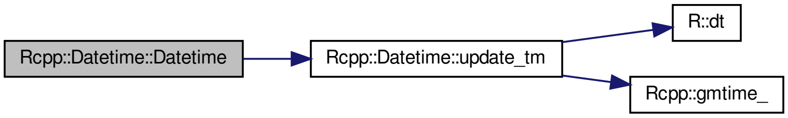
6.172.2.2 Datetime() [2/4]

```
Rcpp::Datetime::Datetime (
             SEXP s )  [inline]
```

Definition at line 25 of file Datetime.h.

References m_dt, and update_tm().

Here is the call graph for this function:



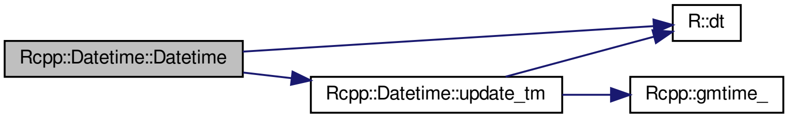
6.172.2.3 Datetime() [3/4]

```
Rcpp::Datetime::Datetime (
             const double & dt )  [inline]
```

Definition at line 42 of file Datetime.h.

References R::dt(), m_dt, and update_tm().

Here is the call graph for this function:



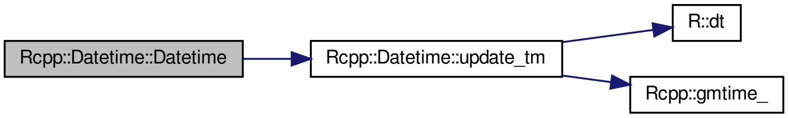
6.172.2.4 Datetime() [4/4]

```
Rcpp::Datetime::Datetime (
             const std::string & s,
             const std::string & fmt = "%Y-%m-%d %H:%M:%OS" )  [inline]
```

Definition at line 30 of file Datetime.h.

References m_dt, and update_tm().

Here is the call graph for this function:



6.172.3 Member Function Documentation

6.172.3.1 baseYear()

```
static unsigned int Rcpp::Datetime::baseYear ( )  [inline], [static], [private]
```

Definition at line 116 of file Datetime.h.

6.172.3.2 format()

```
std::string Rcpp::Datetime::format (
             const char * fmt = "%Y-%m-%d %H:%M:%S" ) const  [inline]
```

Definition at line 75 of file Datetime.h.

References m_dt, m_us, and Rcpp::tm.

6.172.3.3 getDay()

```
int Rcpp::Datetime::getDay ( ) const  [inline]
```

Definition at line 54 of file Datetime.h.

References m_tm.

6.172.3.4 getFractionalTimestamp()

```
double Rcpp::Datetime::getFractionalTimestamp (
             void  ) const  [inline]
```

Definition at line 48 of file Datetime.h.

References m_dt.

Referenced by Rcpp::internal::caster< Rcpp::Datetime, double >(), and Rcpp::wrap< Datetime >().

6.172.3.5 getHours()

```
int Rcpp::Datetime::getHours ( ) const  [inline]
```

Definition at line 53 of file Datetime.h.

References m_tm.

6.172.3.6 getMicroSeconds()

```
int Rcpp::Datetime::getMicroSeconds ( ) const  [inline]
```

Definition at line 50 of file Datetime.h.

References m_us.

6.172.3.7 getMinutes()

```
int Rcpp::Datetime::getMinutes ( ) const  [inline]
```

Definition at line 52 of file Datetime.h.

References m_tm.

6.172.3.8 getMonth()

```
int Rcpp::Datetime::getMonth ( ) const  [inline]
```

Definition at line 55 of file Datetime.h.

References m_tm.

6.172.3.9 getSeconds()

```
int Rcpp::Datetime::getSeconds ( ) const  [inline]
```

Definition at line 51 of file Datetime.h.

References m_tm.

6.172.3.10 getWeekday()

```
int Rcpp::Datetime::getWeekday ( ) const  [inline]
```

Definition at line 57 of file Datetime.h.

References m_tm.

6.172.3.11 getYear()

```
int Rcpp::Datetime::getYear ( ) const  [inline]
```

Definition at line 56 of file Datetime.h.

References m_tm.

6.172.3.12 getYearday()

```
int Rcpp::Datetime::getYearday ( ) const  [inline]
```

Definition at line 58 of file Datetime.h.

References m_tm.

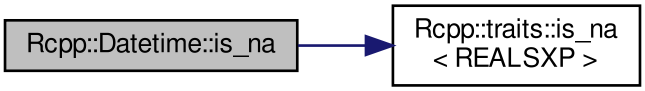
6.172.3.13 is_na()

```
int Rcpp::Datetime::is_na ( ) const  [inline]
```

Definition at line 71 of file Datetime.h.

References Rcpp::traits::is_na< REALSXP >(), and m_dt.

Here is the call graph for this function:



6.172.3.14 operator double()

```
Rcpp::Datetime::operator double ( ) const  [inline]
```

Definition at line 73 of file Datetime.h.

References m_dt.

6.172.3.15 update_tm()

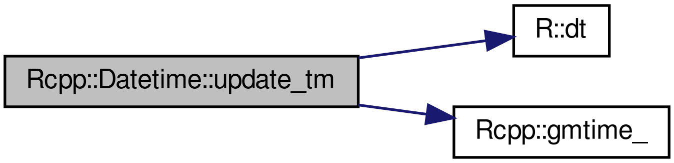
```
void Rcpp::Datetime::update_tm ( )  [inline], [private]
```

Definition at line 100 of file Datetime.h.

References R::dt(), Rcpp::gmtime_(), m_dt, m_tm, and m_us.

Referenced by Datetime().

Here is the call graph for this function:



6.172.4 Friends And Related Function Documentation

6.172.4.1 operator"!=

```
bool operator!= (
            const Datetime & dt1,
            const Datetime & dt2 )  [friend]
```

Definition at line 164 of file Datetime.h.

6.172.4.2 operator+ [1/2]

```
Datetime operator+ (
            const Datetime & dt,
            double offset )  [friend]
```

Definition at line 138 of file Datetime.h.

6.172.4.3 operator+ [2/2]

```
Datetime operator+ (
             const Datetime & dt,
             int offset )  [friend]
```

Definition at line 148 of file Datetime.h.

6.172.4.4 operator-

```
double operator- (
             const Datetime & dt1,
             const Datetime & dt2 )  [friend]
```

Definition at line 158 of file Datetime.h.

6.172.4.5 operator<

```
bool operator< (
             const Datetime & dt1,
             const Datetime & dt2 )  [friend]
```

Definition at line 159 of file Datetime.h.

6.172.4.6 operator<<

```
std::ostream& operator<< (
             std::ostream & s,
             const Datetime d )  [friend]
```

Definition at line 166 of file Datetime.h.

6.172.4.7 operator<=

```
bool operator<= (
             const Datetime & dt1,
             const Datetime & dt2 )  [friend]
```

Definition at line 163 of file Datetime.h.

6.172.4.8 operator==

```
bool operator== (
            const Datetime & dt1,
            const Datetime & dt2 )  [friend]
```

Definition at line 161 of file Datetime.h.

6.172.4.9 operator>

```
bool operator> (
            const Datetime & dt1,
            const Datetime & dt2 )  [friend]
```

Definition at line 160 of file Datetime.h.

6.172.4.10 operator>=

```
bool operator>= (
            const Datetime & dt1,
            const Datetime & dt2 )  [friend]
```

Definition at line 162 of file Datetime.h.

6.172.5 Member Data Documentation

6.172.5.1 m_dt

```
double Rcpp::Datetime::m_dt  [private]
```

Definition at line 95 of file Datetime.h.

Referenced by Datetime(), format(), getFractionalTimestamp(), is_na(), operator double(), and update_tm().

6.172.5.2 m_tm

```
struct tm Rcpp::Datetime::m_tm  [private]
```

Definition at line 95 of file Datetime.h.

Referenced by getDay(), getHours(), getMinutes(), getMonth(), getSeconds(), getWeekday(), getYear(), getYearday(), and update_tm().

6.172.5.3 m_us

```
unsigned int Rcpp::Datetime::m_us  [private]
```

Definition at line 97 of file Datetime.h.

Referenced by format(), getMicroSeconds(), and update_tm().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/Datetime.h

6.173 Rcpp::algorithm::helpers::decays_to_ctype< T > Struct Template Reference

```
#include <algorithm.h>
```

Static Public Member Functions

- static CTYPE_CHAR test (const char &)
- static CTYPE_SHORT test (const short &)
- static CTYPE_INT test (const int &)
- static CTYPE_LONG test (const long &)
- static CTYPE_FLOAT test (const float &)
- static CTYPE_DOUBLE test (const double &)
- static CTYPE_LONG_DOUBLE test (const long double &)
- static CTYPE_STRING test (const std::string &)
- static CTYPE_UNSIGNED_CHAR test (const unsigned char &)
- static CTYPE_UNSIGNED_SHORT test (const unsigned short &)
- static CTYPE_UNSIGNED_INT test (const unsigned int &)
- static CTYPE_UNSIGNED_LONG test (const unsigned long &)
- static CTYPE_UNKNOWN test (...)
- static T make ()

Static Public Attributes

- static const bool value = ctype_helper< sizeof(test(make())) >::value

6.173.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::decays_to_ctype< T >

Definition at line 136 of file algorithm.h.

6.173.2 Member Function Documentation

6.173.2.1 make()

```
template<typename T >
static T Rcpp::algorithm::helpers::decays_to_ctype< T >::make ( )  [static]
```

6.173.2.2 test() [1/13]

```
template<typename T >
static CTYPE_UNKNOWN Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
              ... )  [static]
```

6.173.2.3 test() [2/13]

```
template<typename T >
static CTYPE_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const char & )  [static]
```

6.173.2.4 test() [3/13]

```
template<typename T >
static CTYPE_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const double & )  [static]
```

6.173.2.5 test() [4/13]

```
template<typename T >
static CTYPE_FLOAT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const float & ) [static]
```

6.173.2.6 test() [5/13]

```
template<typename T >
static CTYPE_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const int & ) [static]
```

6.173.2.7 test() [6/13]

```
template<typename T >
static CTYPE_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const long & ) [static]
```

6.173.2.8 test() [7/13]

```
template<typename T >
static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const long double & ) [static]
```

6.173.2.9 test() [8/13]

```
template<typename T >
static CTYPE_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const short & ) [static]
```

6.173.2.10 test() [9/13]

```
template<typename T >
static CTYPE_STRING Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const std::string & ) [static]
```

6.173.2.11 test() [10/13]

```
template<typename T >
static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const unsigned char & )  [static]
```

6.173.2.12 test() [11/13]

```
template<typename T >
static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const unsigned int & )  [static]
```

6.173.2.13 test() [12/13]

```
template<typename T >
static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const unsigned long & )  [static]
```

6.173.2.14 test() [13/13]

```
template<typename T >
static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
             const unsigned short & )  [static]
```

6.173.3 Member Data Documentation

6.173.3.1 value

```
template<typename T >
const bool Rcpp::algorithm::helpers::decays_to_ctype< T >::value = ctype_helper< sizeof(test(make()))
>::value  [static]
```

Definition at line 160 of file algorithm.h.

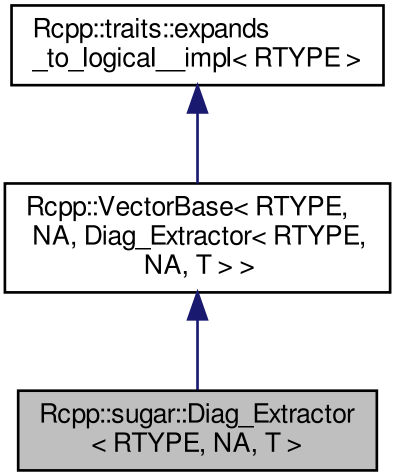
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.174 Rcpp::sugar::Diag_Extractor< RTYPE, NA, T > Class Template Reference

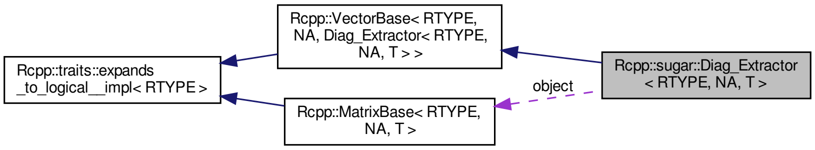
```
#include <diag.h>
```

Inheritance diagram for Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >:



Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Diag_Extractor< RTYPE, NA, T > >
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >:



Rcpp::VectorBase< RTYPE, NA, Diag_Extractor< RTYPE, NA, T > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >
object
Rcpp::MatrixBase< RTYPE, NA, T >

Public Types

- typedef Rcpp::MatrixBase< RTYPE, NA, T > MAT_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Diag_Extractor (const MAT_TYPE &object_)
- STORAGE operator[ ] (int i) const
- R_xlen_t size () const

Private Attributes

- const MAT_TYPE & object
- R_xlen_t n

6.174.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >

Definition at line 29 of file diag.h.

6.174.2 Member Typedef Documentation

6.174.2.1 MAT_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::MatrixBase<RTYPE,NA,T> Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::MAT_TYPE
```

Definition at line 31 of file diag.h.

6.174.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >↩
::STORAGE
```

Definition at line 32 of file diag.h.

6.174.3 Constructor & Destructor Documentation

6.174.3.1 Diag_Extractor()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::Diag_Extractor (
             const MAT_TYPE & object_ )  [inline]
```

Definition at line 34 of file diag.h.

References Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::n.

6.174.4 Member Function Documentation

6.174.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 40 of file diag.h.

References Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::object.

6.174.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 43 of file diag.h.

References Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::n.

6.174.5 Member Data Documentation

6.174.5.1 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::n  [private]
```

Definition at line 47 of file diag.h.

Referenced by Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::Diag_Extractor(), and Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::size().

6.174.5.2 object

```
template<int RTYPE, bool NA, typename T >
const MAT_TYPE& Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::object  [private]
```

Definition at line 46 of file diag.h.

Referenced by Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::operator[ ]().

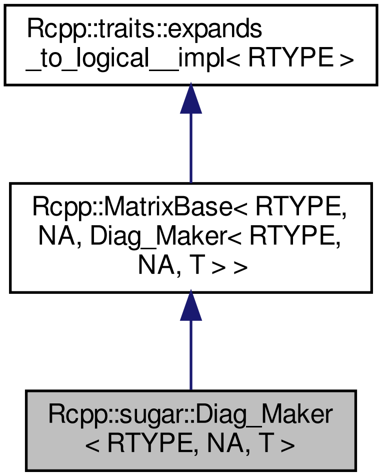
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

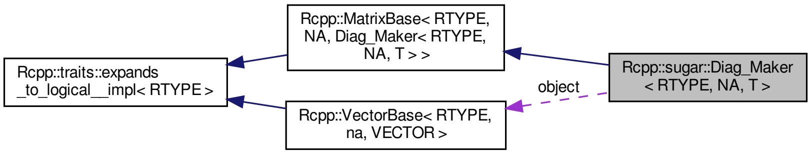
6.175 Rcpp::sugar::Diag_Maker< RTYPE, NA, T > Class Template Reference

```
#include <diag.h>
```

Inheritance diagram for Rcpp::sugar::Diag_Maker< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Diag_Maker< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Diag_Maker (const VEC_TYPE &object_)
- STORAGE operator() (int i, int j) const
- R_xlen_t size () const
- int ncol () const
- int nrow () const

Private Attributes

- const VEC_TYPE & object
- int n

Additional Inherited Members

6.175.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Diag_Maker< RTYPE, NA, T >

Definition at line 52 of file diag.h.

6.175.2 Member Typedef Documentation

6.175.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::STORAGE
```

Definition at line 55 of file diag.h.

6.175.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 54 of file diag.h.

6.175.3 Constructor & Destructor Documentation

6.175.3.1 Diag_Maker()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::Diag_Maker (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 57 of file diag.h.

6.175.4 Member Function Documentation

6.175.4.1 ncol()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::ncol ( ) const  [inline]
```

Definition at line 63 of file diag.h.

References Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::n.

6.175.4.2 nrow()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::nrow ( ) const  [inline]
```

Definition at line 64 of file diag.h.

References Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::n.

6.175.4.3 operator()()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::operator() (
             int i,
             int j ) const [inline]
```

Definition at line 59 of file diag.h.

6.175.4.4 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 62 of file diag.h.

References Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::n.

6.175.5 Member Data Documentation

6.175.5.1 n

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::n [private]
```

Definition at line 68 of file diag.h.

Referenced by Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::ncol(), Rcpp::sugar::Diag_Maker< RTYPE, NA, T >↩ ::nrow(), and Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::size().

6.175.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::object [private]
```

Definition at line 67 of file diag.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

6.176 Rcpp::sugar::diag_result_type_trait< T > Struct Template Reference

```
#include <diag.h>
```

Public Types

- typedef Rcpp::traits::if_< Rcpp::traits::matrix_interface< T >::value, Diag_Extractor< T::r_type::value, T::can_have_na::value, T >, Diag_Maker< T::r_type::value, T::can_have_na::value, T > >::type type

6.176.1 Detailed Description

template<typename T>
struct Rcpp::sugar::diag_result_type_trait< T >

Definition at line 71 of file diag.h.

6.176.2 Member Typedef Documentation

6.176.2.1 type

```
template<typename T >
typedef Rcpp::traits::if_< Rcpp::traits::matrix_interface<T>::value, Diag_Extractor< T::r_type↩
::value , T::can_have_na::value , T >, Diag_Maker< T::r_type::value , T::can_have_na::value , T >
>::type Rcpp::sugar::diag_result_type_trait< T >::type
```

Definition at line 76 of file diag.h.

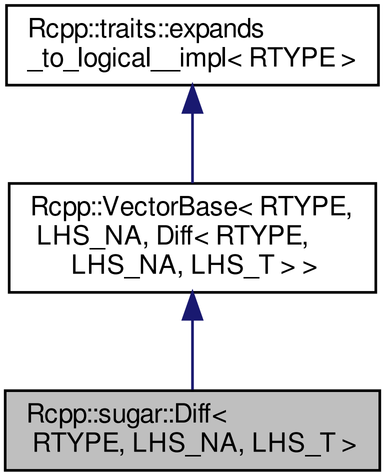
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

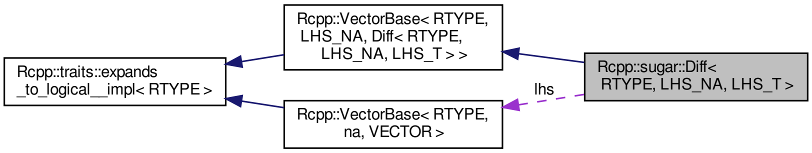
6.177 Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > >
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >

Collaboration diagram for Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Diff (const LHS_TYPE &lhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- void set_previous (R_xlen_t i, STORAGE value) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- STORAGE previous
- R_xlen_t previous_index
- bool was_na

6.177.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >

Definition at line 32 of file diff.h.

6.177.2 Member Typedef Documentation

6.177.2.1 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 34 of file diff.h.

6.177.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >↩
::STORAGE
```

Definition at line 35 of file diff.h.

6.177.3 Constructor & Destructor Documentation

6.177.3.1 Diff()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::Diff (
            const LHS_TYPE & lhs_ )  [inline]
```

Definition at line 37 of file diff.h.

6.177.4 Member Function Documentation

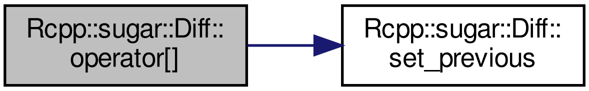
6.177.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 44 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous(), and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::was_na.

Here is the call graph for this function:



6.177.4.2 set_previous()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
void Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous (
             R_xlen_t i,
             STORAGE value ) const  [inline]
```

Definition at line 59 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index, and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::was_na.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[ ]().

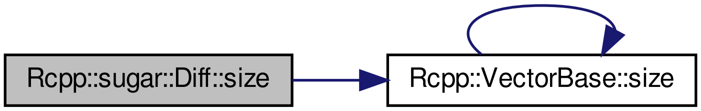
6.177.4.3 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 65 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >↩ ::size().

Here is the call graph for this function:



6.177.5 Member Data Documentation

6.177.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 68 of file diff.h.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[ ](), Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size(), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size(), and Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size().

6.177.5.2 previous

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous  [mutable], [private]
```

Definition at line 69 of file diff.h.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[ ](), and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous().

6.177.5.3 previous_index

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index  [mutable], [private]
```

Definition at line 70 of file diff.h.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[ ](), and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous().

6.177.5.4 was_na

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::was_na  [mutable], [private]
```

Definition at line 71 of file diff.h.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[ ](), and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous().

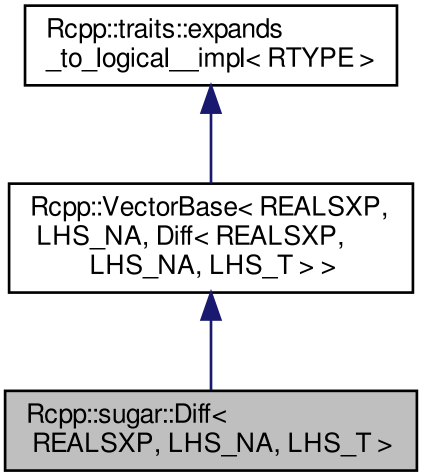
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/diff.h

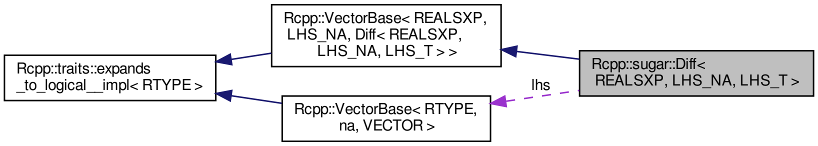
6.178 Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > >
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >

Collaboration diagram for Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE

Public Member Functions

- Diff (const LHS_TYPE &lhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- double previous
- R_xlen_t previous_index

6.178.1 Detailed Description

template<typename LHS_T, bool LHS_NA>
class Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >

Definition at line 75 of file diff.h.

6.178.2 Member Typedef Documentation

6.178.2.1 LHS_TYPE

```
template<typename LHS_T , bool LHS_NA>
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >↩
::LHS_TYPE
```

Definition at line 77 of file diff.h.

6.178.3 Constructor & Destructor Documentation

6.178.3.1 Diff()

```
template<typename LHS_T , bool LHS_NA>
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::Diff (
             const LHS_TYPE & lhs_ )  [inline]
```

Definition at line 79 of file diff.h.

6.178.4 Member Function Documentation

6.178.4.1 operator[]()

```
template<typename LHS_T , bool LHS_NA>
double Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 81 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >↩
::previous, and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index.

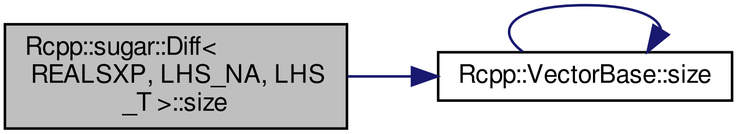
6.178.4.2 size()

```
template<typename LHS_T , bool LHS_NA>
R_xlen_t Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 89 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >↩
::size().

Here is the call graph for this function:



6.178.5 Member Data Documentation

6.178.5.1 lhs

```
template<typename LHS_T , bool LHS_NA>
const LHS_TYPE& Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 92 of file diff.h.

6.178.5.2 previous

```
template<typename LHS_T , bool LHS_NA>
double Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::previous  [mutable], [private]
```

Definition at line 93 of file diff.h.

6.178.5.3 previous_index

```
template<typename LHS_T , bool LHS_NA>
R_xlen_t Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::previous_index  [mutable], [private]
```

Definition at line 94 of file diff.h.

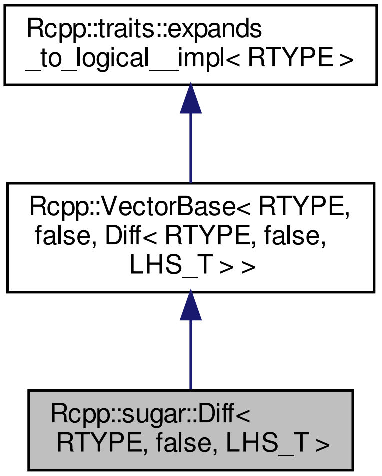
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/diff.h

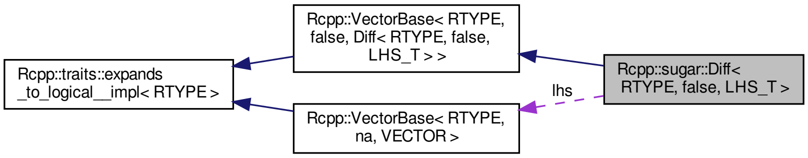
6.179 Rcpp::sugar::Diff< RTYPE, false, LHS_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< RTYPE, false, LHS_T >:



Collaboration diagram for Rcpp::sugar::Diff< RTYPE, false, LHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Diff (const LHS_TYPE &lhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- STORAGE previous
- R_xlen_t previous_index

6.179.1 Detailed Description

template<int RTYPE, typename LHS_T>
class Rcpp::sugar::Diff< RTYPE, false, LHS_T >

Definition at line 98 of file diff.h.

6.179.2 Member Typedef Documentation

6.179.2.1 LHS_TYPE

```
template<int RTYPE, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Diff< RTYPE, false, LHS_T >::LHS_TYPE
```

Definition at line 100 of file diff.h.

6.179.2.2 STORAGE

```
template<int RTYPE, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diff< RTYPE, false, LHS_T >::STORAGE
```

Definition at line 101 of file diff.h.

6.179.3 Constructor & Destructor Documentation

6.179.3.1 Diff()

```
template<int RTYPE, typename LHS_T >
Rcpp::sugar::Diff< RTYPE, false, LHS_T >::Diff (
             const LHS_TYPE & lhs_ )  [inline]
```

Definition at line 103 of file diff.h.

6.179.4 Member Function Documentation

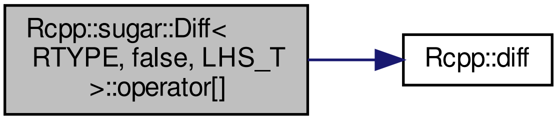
6.179.4.1 operator[]()

```
template<int RTYPE, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 105 of file diff.h.

References Rcpp::diff(), Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous, and Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index.

Here is the call graph for this function:



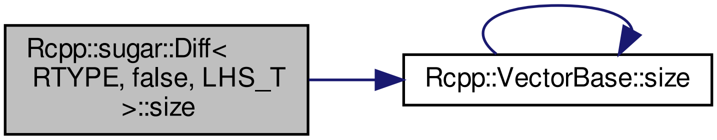
6.179.4.2 size()

```
template<int RTYPE, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size ( ) const  [inline]
```

Definition at line 113 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.179.5 Member Data Documentation

6.179.5.1 lhs

```
template<int RTYPE, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Diff< RTYPE, false, LHS_T >::lhs  [private]
```

Definition at line 116 of file diff.h.

6.179.5.2 previous

```
template<int RTYPE, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, false, LHS_T >::previous  [mutable], [private]
```

Definition at line 117 of file diff.h.

6.179.5.3 previous_index

```
template<int RTYPE, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, false, LHS_T >::previous_index  [mutable], [private]
```

Definition at line 118 of file diff.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/diff.h

6.180 Rcpp::Dimension Class Reference

```
#include <Dimension.h>
```

Public Types

- typedef std::vector< int >::reference reference
- typedef std::vector< int >::const_reference const_reference

Public Member Functions

- Dimension ()
- Dimension (SEXP dims)
- Dimension (const Dimension &other)
- Dimension & operator= (const Dimension &other)
- Dimension (const size_t &n1)
- Dimension (const size_t &n1, const size_t &n2)
- Dimension (const size_t &n1, const size_t &n2, const size_t &n3)
- operator SEXP () const
- int size () const
- R_xlen_t prod () const
- reference operator[ ] (int i)
- const_reference operator[ ] (int i) const

Private Attributes

- std::vector< int > dims

6.180.1 Detailed Description

Definition at line 27 of file Dimension.h.

6.180.2 Member Typedef Documentation

6.180.2.1 const_reference

```
typedef std::vector<int>::const_reference Rcpp::Dimension::const_reference
```

Definition at line 30 of file Dimension.h.

6.180.2.2 reference

```
typedef std::vector<int>::reference Rcpp::Dimension::reference
```

Definition at line 29 of file Dimension.h.

6.180.3 Constructor & Destructor Documentation

6.180.3.1 Dimension() [1/6]

```
Rcpp::Dimension::Dimension ( )  [inline]
```

Definition at line 32 of file Dimension.h.

6.180.3.2 Dimension() [2/6]

```
Rcpp::Dimension::Dimension (
             SEXP dims )  [inline]
```

Definition at line 23 of file Dimension.h.

6.180.3.3 Dimension() [3/6]

```
Rcpp::Dimension::Dimension (
             const Dimension & other )  [inline]
```

Definition at line 36 of file Dimension.h.

6.180.3.4 Dimension() [4/6]

```
Rcpp::Dimension::Dimension (
             const size_t & n1 )  [inline]
```

Definition at line 42 of file Dimension.h.

References dims.

6.180.3.5 Dimension() [5/6]

```
Rcpp::Dimension::Dimension (
             const size_t & n1,
             const size_t & n2 )  [inline]
```

Definition at line 45 of file Dimension.h.

References dims.

6.180.3.6 Dimension() [6/6]

```
Rcpp::Dimension::Dimension (
             const size_t & n1,
             const size_t & n2,
             const size_t & n3 )  [inline]
```

Definition at line 49 of file Dimension.h.

References dims.

6.180.4 Member Function Documentation

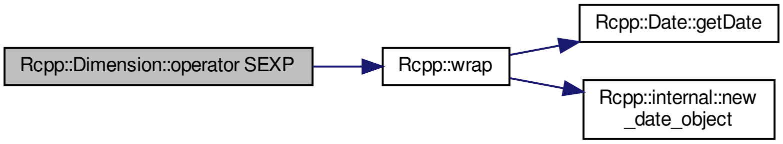
6.180.4.1 operator SEXP()

```
Rcpp::Dimension::operator SEXP ( ) const  [inline]
```

Definition at line 25 of file Dimension.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.180.4.2 operator=()

```
Dimension& Rcpp::Dimension::operator= (
             const Dimension & other )  [inline]
```

Definition at line 37 of file Dimension.h.

References dims.

6.180.4.3 operator[]() [1/2]

```
reference Rcpp::Dimension::operator[] (
            int i ) [inline]
```

Definition at line 63 of file Dimension.h.

References dims.

6.180.4.4 operator[]() [2/2]

```
const_reference Rcpp::Dimension::operator[] (
            int i ) const [inline]
```

Definition at line 67 of file Dimension.h.

References dims.

6.180.4.5 prod()

```
R_xlen_t Rcpp::Dimension::prod ( ) const [inline]
```

Definition at line 59 of file Dimension.h.

References dims.

6.180.4.6 size()

```
int Rcpp::Dimension::size ( ) const [inline]
```

Definition at line 56 of file Dimension.h.

References dims.

6.180.5 Member Data Documentation

6.180.5.1 dims

```
std::vector<int> Rcpp::Dimension::dims  [private]
```

Definition at line 73 of file Dimension.h.

Referenced by Dimension(), operator=(), operator[ ](), prod(), and size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Dimension.h

6.181 Rcpp::internal::DimNameProxy Class Reference

```
#include <DimNameProxy.h>
```

Public Member Functions

- DimNameProxy (SEXP data, int dim)
- DimNameProxy (DimNameProxy const &other)
- DimNameProxy & assign (SEXP other)
- DimNameProxy & operator= (SEXP other)
- DimNameProxy & operator= (const DimNameProxy &other)
- operator SEXP () const
- template<typename T >
  operator T () const

Private Attributes

- SEXP data_
- int dim_

6.181.1 Detailed Description

Definition at line 28 of file DimNameProxy.h.

6.181.2 Constructor & Destructor Documentation

6.181.2.1 DimNameProxy() [1/2]

```
Rcpp::internal::DimNameProxy::DimNameProxy (
             SEXP data,
             int dim )  [inline]
```

Definition at line 32 of file DimNameProxy.h.

6.181.2.2 DimNameProxy() [2/2]

```
Rcpp::internal::DimNameProxy::DimNameProxy (
             DimNameProxy const & other )  [inline]
```

Definition at line 33 of file DimNameProxy.h.

6.181.3 Member Function Documentation

6.181.3.1 assign()

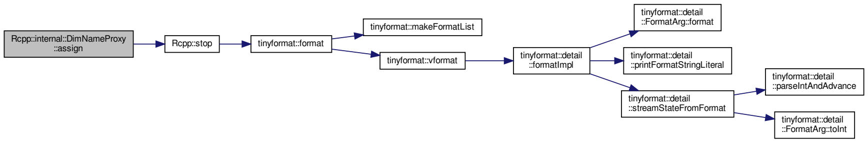
```
DimNameProxy& Rcpp::internal::DimNameProxy::assign (
             SEXP other )  [inline]
```

Definition at line 36 of file DimNameProxy.h.

References data_, dim_, and Rcpp::stop().

Referenced by operator=().

Here is the call graph for this function:


Rcpp::internal::DimNameProxy ::assign
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.181.3.2 operator SEXP()

```
Rcpp::internal::DimNameProxy::operator SEXP ( ) const  [inline]
```

Definition at line 66 of file DimNameProxy.h.

References data_, and dim_.

6.181.3.3 operator T()

```
template<typename T >
Rcpp::internal::DimNameProxy::operator T ( ) const  [inline]
```

Definition at line 72 of file DimNameProxy.h.

References data_, and dim_.

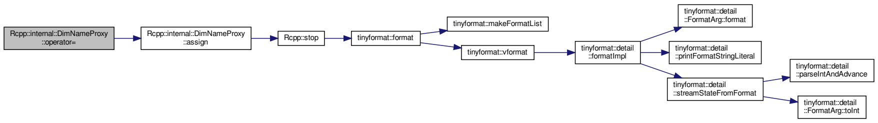
6.181.3.4 operator=() [1/2]

```
DimNameProxy& Rcpp::internal::DimNameProxy::operator= (
             const DimNameProxy & other )  [inline]
```

Definition at line 62 of file DimNameProxy.h.

References assign().

Here is the call graph for this function:


Rcpp::internal::DimNameProxy ::operator=
Rcpp::internal::DimNameProxy ::assign
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

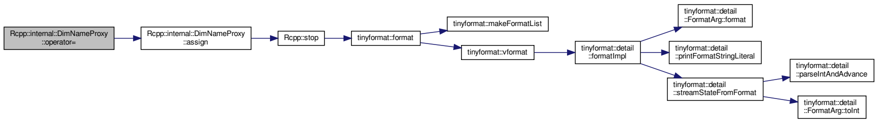
6.181.3.5 operator=() [2/2]

```
DimNameProxy& Rcpp::internal::DimNameProxy::operator= (
             SEXP other )  [inline]
```

Definition at line 58 of file DimNameProxy.h.

References assign().

Here is the call graph for this function:


Rcpp::internal::DimNameProxy ::operator=
Rcpp::internal::DimNameProxy ::assign
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.181.4 Member Data Documentation

6.181.4.1 data_

```
SEXP Rcpp::internal::DimNameProxy::data_  [private]
```

Definition at line 83 of file DimNameProxy.h.

Referenced by assign(), operator SEXP(), and operator T().

6.181.4.2 dim_

```
int Rcpp::internal::DimNameProxy::dim_  [private]
```

Definition at line 84 of file DimNameProxy.h.

Referenced by assign(), operator SEXP(), and operator T().

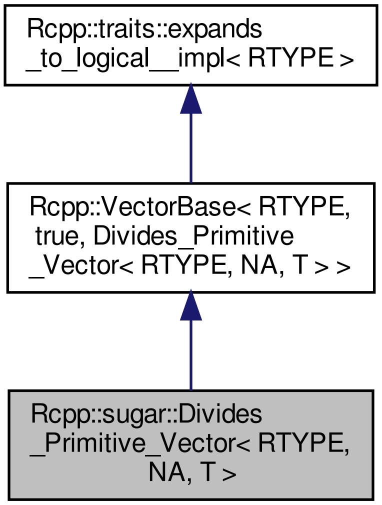
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/DimNameProxy.h

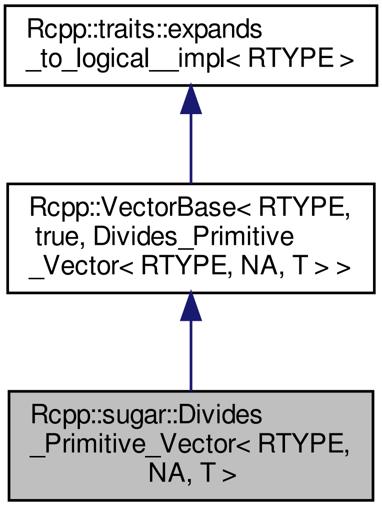
6.182 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Divides_Primitive _Vector< RTYPE, NA, T > >
Rcpp::sugar::Divides _Primitive_Vector< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type VEC_EXT
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Divides_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- STORAGE lhs
- const VEC_EXT & rhs
- bool lhs_na

6.182.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >

Definition at line 322 of file divides.h.

6.182.2 Member Typedef Documentation

6.182.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T
>::STORAGE
```

Definition at line 327 of file divides.h.

6.182.2.2 VEC_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE,
NA, T >::VEC_EXT
```

Definition at line 326 of file divides.h.

6.182.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >↩
::VEC_TYPE
```

Definition at line 325 of file divides.h.

6.182.3 Constructor & Destructor Documentation

6.182.3.1 Divides_Primitive_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::Divides_Primitive_Vector (
             STORAGE lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 329 of file divides.h.

6.182.4 Member Function Documentation

6.182.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 332 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs, Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs_na, and Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.182.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 337 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.182.5 Member Data Documentation

6.182.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 339 of file divides.h.

Referenced by Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::operator[](), Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::operator[](), Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::operator[](), and Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::operator[]().

6.182.5.2 lhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs_na  [private]
```

Definition at line 341 of file divides.h.

Referenced by Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Divides_↩ Primitive_Vector< RTYPE, false, T >::operator[ ]().

6.182.5.3 rhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs  [private]
```

Definition at line 340 of file divides.h.

Referenced by Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Divides_↩ Primitive_Vector< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >↩ ::operator[ ](), Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Divides↩ _Primitive_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::size(), Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::size(), and Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::size().

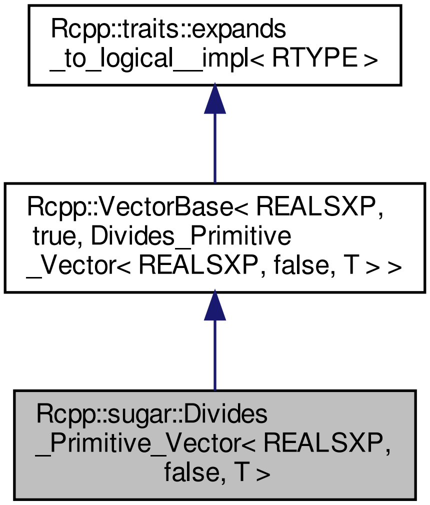
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

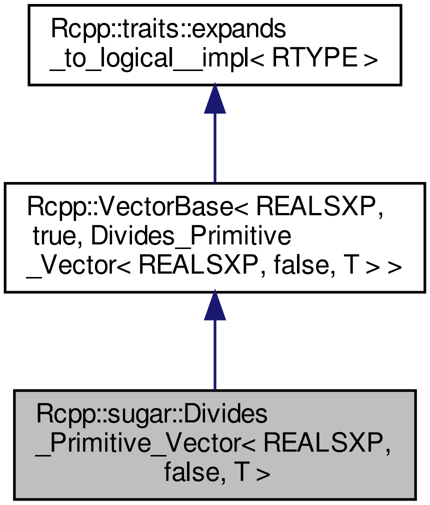
6.183 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Primitive
_Vector< REALSXP, false, T > >
Rcpp::sugar::Divides
_Primitive_Vector< REALSXP,
false, T >

Collaboration diagram for Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Primitive
_Vector< REALSXP, false, T > >
Rcpp::sugar::Divides
_Primitive_Vector< REALSXP,
false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type VEC_EXT

Public Member Functions

- Divides_Primitive_Vector (double lhs_, const VEC_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- double lhs
- const VEC_EXT & rhs

6.183.1 Detailed Description

template<typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >

Definition at line 389 of file divides.h.

6.183.2 Member Typedef Documentation

6.183.2.1 VEC_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Divides_Primitive_Vector<
REALSXP, false, T >::VEC_EXT
```

Definition at line 393 of file divides.h.

6.183.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 392 of file divides.h.

6.183.3 Constructor & Destructor Documentation

6.183.3.1 Divides_Primitive_Vector()

```
template<typename T >
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::Divides_Primitive_Vector (
             double lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 395 of file divides.h.

6.183.4 Member Function Documentation

6.183.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 398 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.183.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 401 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.183.5 Member Data Documentation

6.183.5.1 lhs

```
template<typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::lhs  [private]
```

Definition at line 404 of file divides.h.

6.183.5.2 rhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::rhs  [private]
```

Definition at line 405 of file divides.h.

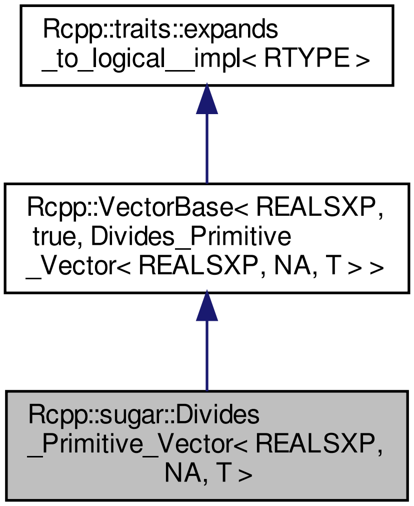
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

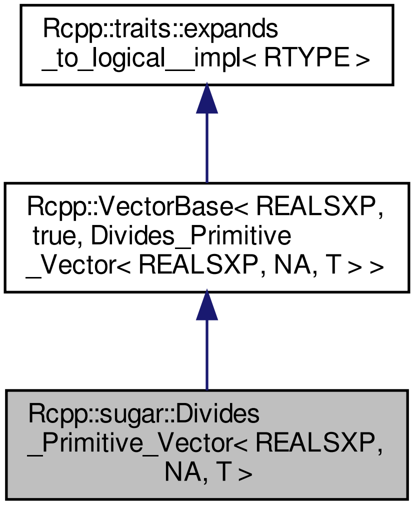
6.184 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Primitive
_Vector< REALSXP, NA, T > >
Rcpp::sugar::Divides
_Primitive_Vector< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Primitive
_Vector< REALSXP, NA, T > >
Rcpp::sugar::Divides
_Primitive_Vector< REALSXP,
NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type VEC_EXT

Public Member Functions

- Divides_Primitive_Vector (double lhs_, const VEC_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- double lhs
- const VEC_EXT & rhs

6.184.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >

Definition at line 345 of file divides.h.

6.184.2 Member Typedef Documentation

6.184.2.1 VEC_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Divides_Primitive_Vector< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 349 of file divides.h.

6.184.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 348 of file divides.h.

6.184.3 Constructor & Destructor Documentation

6.184.3.1 Divides_Primitive_Vector()

```
template<bool NA, typename T >
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::Divides_Primitive_Vector (
             double lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 351 of file divides.h.

6.184.4 Member Function Documentation

6.184.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 354 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::Divides_Primitive_↩ Vector< RTYPE, NA, T >::rhs.

6.184.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 357 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.184.5 Member Data Documentation

6.184.5.1 lhs

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::lhs  [private]
```

Definition at line 359 of file divides.h.

6.184.5.2 rhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::rhs  [private]
```

Definition at line 360 of file divides.h.

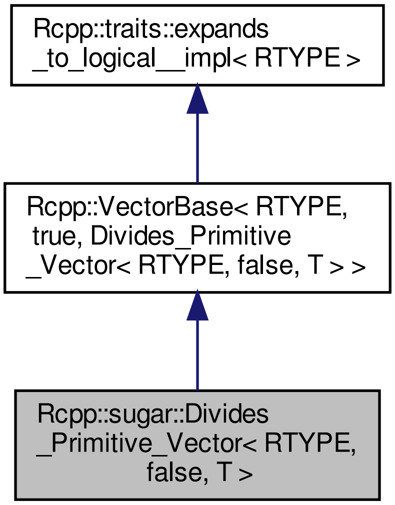
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

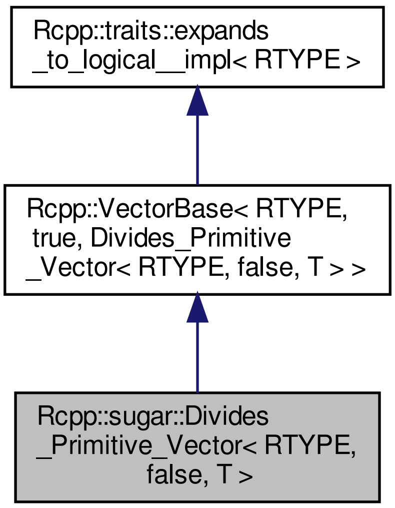
6.185 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type VEC_EXT

Public Member Functions

- Divides_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- STORAGE lhs
- const VEC_EXT & rhs
- bool lhs_na

6.185.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >

Definition at line 366 of file divides.h.

6.185.2 Member Typedef Documentation

6.185.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T
>::STORAGE
```

Definition at line 370 of file divides.h.

6.185.2.2 VEC_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE,false,T>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE,
false, T >::VEC_EXT
```

Definition at line 371 of file divides.h.

6.185.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 369 of file divides.h.

6.185.3 Constructor & Destructor Documentation

6.185.3.1 Divides_Primitive_Vector()

```
template<int RTYPE, typename T >
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::Divides_Primitive_Vector (
             STORAGE lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 373 of file divides.h.

6.185.4 Member Function Documentation

6.185.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 376 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs, Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs_na, and Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.185.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 380 of file divides.h.

References Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs.

6.185.5 Member Data Documentation

6.185.5.1 lhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::lhs  [private]
```

Definition at line 383 of file divides.h.

6.185.5.2 lhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::lhs_na  [private]
```

Definition at line 385 of file divides.h.

6.185.5.3 rhs

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::rhs  [private]
```

Definition at line 384 of file divides.h.

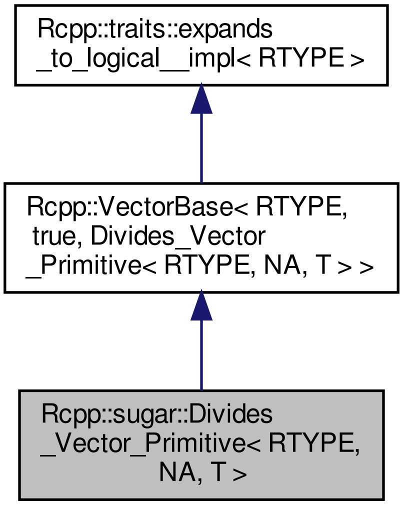
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

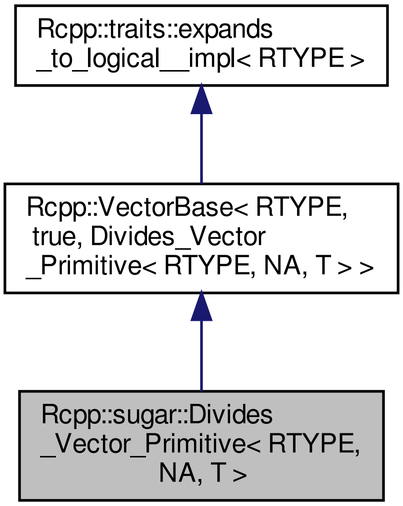
6.186 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >:



Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type VEC_EXT

Public Member Functions

- Divides_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- STORAGE rhs
- bool rhs_na

6.186.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >

Definition at line 226 of file divides.h.

6.186.2 Member Typedef Documentation

6.186.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T
>::STORAGE
```

Definition at line 229 of file divides.h.

6.186.2.2 VEC_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE,
NA, T >::VEC_EXT
```

Definition at line 231 of file divides.h.

6.186.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >↩
::VEC_TYPE
```

Definition at line 230 of file divides.h.

6.186.3 Constructor & Destructor Documentation

6.186.3.1 Divides_Vector_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::Divides_Vector_Primitive (
                const VEC_TYPE & lhs_,
                STORAGE rhs_ )  [inline]
```

Definition at line 233 of file divides.h.

6.186.4 Member Function Documentation

6.186.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 237 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.186.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 243 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs.

6.186.5 Member Data Documentation

6.186.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs  [private]
```

Definition at line 246 of file divides.h.

Referenced by Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::size(), Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::size(), Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::size(), and Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::size().

6.186.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs  [private]
```

Definition at line 247 of file divides.h.

Referenced by Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::operator[ ]().

6.186.5.3 rhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na  [private]
```

Definition at line 248 of file divides.h.

Referenced by Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[ ]().

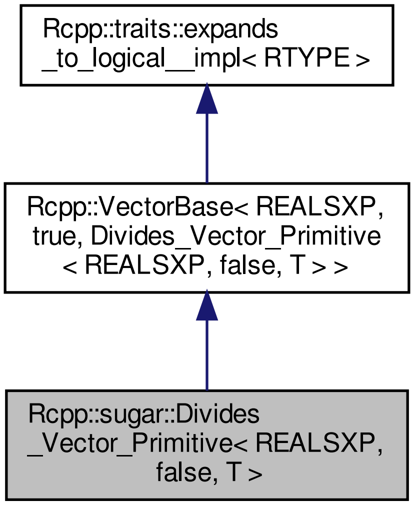
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

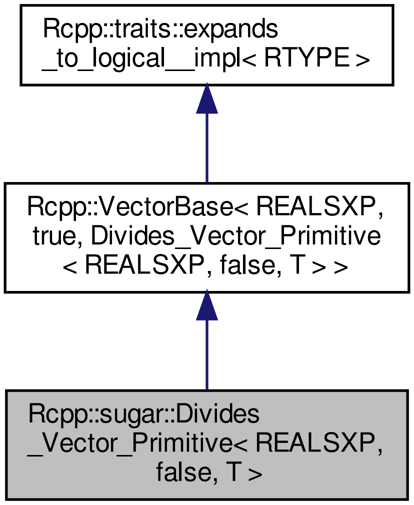
6.187 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, false, T > >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, false, T > >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type VEC_EXT

Public Member Functions

- Divides_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- double rhs

6.187.1 Detailed Description

template<typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >

Definition at line 300 of file divides.h.

6.187.2 Member Typedef Documentation

6.187.2.1 VEC_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Divides_Vector_Primitive<
REALSXP, false, T >::VEC_EXT
```

Definition at line 304 of file divides.h.

6.187.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 303 of file divides.h.

6.187.3 Constructor & Destructor Documentation

6.187.3.1 Divides_Vector_Primitive()

```
template<typename T >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::Divides_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 306 of file divides.h.

6.187.4 Member Function Documentation

6.187.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 309 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Divides_Vector_↩Primitive< RTYPE, NA, T >::rhs.

6.187.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 312 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs.

6.187.5 Member Data Documentation

6.187.5.1 lhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::lhs  [private]
```

Definition at line 315 of file divides.h.

6.187.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::rhs  [private]
```

Definition at line 316 of file divides.h.

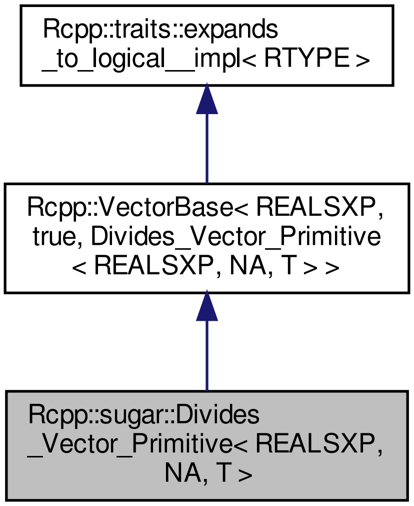
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

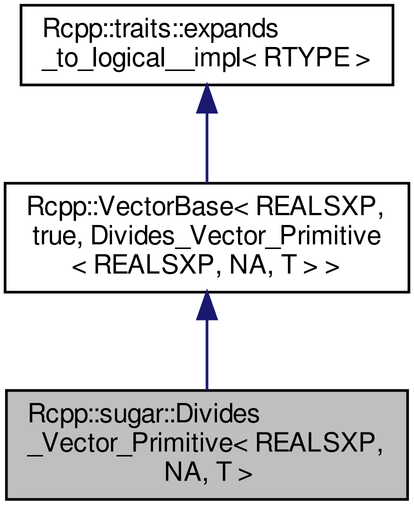
6.188 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Divides
_Vector_Primitive< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Divides_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Divides
_Vector_Primitive< REALSXP,
NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type VEC_EXT

Public Member Functions

- Divides_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- double rhs

6.188.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >

Definition at line 252 of file divides.h.

6.188.2 Member Typedef Documentation

6.188.2.1 VEC_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Divides_Vector_Primitive< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 256 of file divides.h.

6.188.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 255 of file divides.h.

6.188.3 Constructor & Destructor Documentation

6.188.3.1 Divides_Vector_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::Divides_Vector_Primitive (
            const VEC_TYPE & lhs_,
            double rhs_ )  [inline]
```

Definition at line 258 of file divides.h.

6.188.4 Member Function Documentation

6.188.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 262 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Divides_Vector_↩
Primitive< RTYPE, NA, T >::rhs.

6.188.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 266 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs.

6.188.5 Member Data Documentation

6.188.5.1 lhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::lhs  [private]
```

Definition at line 269 of file divides.h.

6.188.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::rhs  [private]
```

Definition at line 270 of file divides.h.

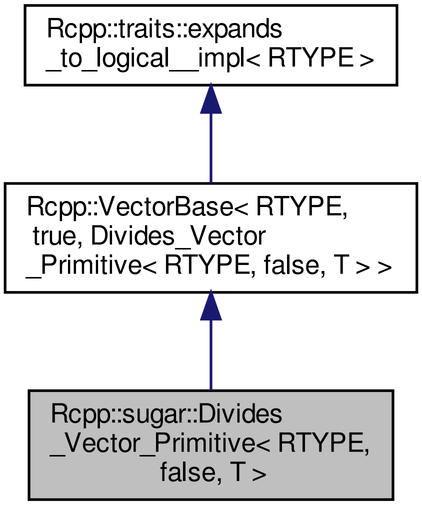
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

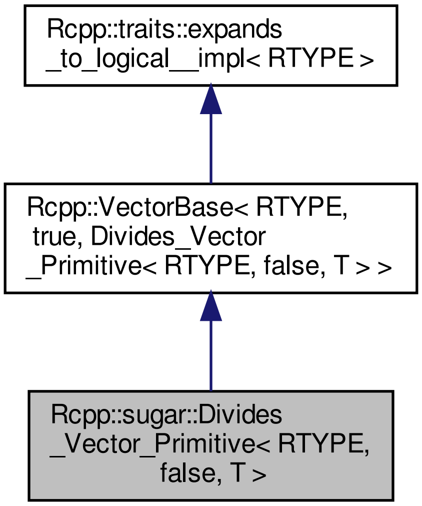
6.189 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Divides_Vector _Primitive< RTYPE, false, T > >
Rcpp::sugar::Divides _Vector_Primitive< RTYPE, false, T >

Collaboration diagram for Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >:



Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type VEC_EXT

Public Member Functions

- Divides_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- STORAGE rhs
- bool rhs_na

6.189.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >

Definition at line 276 of file divides.h.

6.189.2 Member Typedef Documentation

6.189.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 279 of file divides.h.

6.189.2.2 VEC_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE,false,T>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE,
false, T >::VEC_EXT
```

Definition at line 281 of file divides.h.

6.189.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 280 of file divides.h.

6.189.3 Constructor & Destructor Documentation

6.189.3.1 Divides_Vector_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::Divides_Vector_Primitive (
                const VEC_TYPE & lhs_,
                STORAGE rhs_ )  [inline]
```

Definition at line 283 of file divides.h.

6.189.4 Member Function Documentation

6.189.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 286 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.189.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 291 of file divides.h.

References Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs.

6.189.5 Member Data Documentation

6.189.5.1 lhs

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::lhs  [private]
```

Definition at line 294 of file divides.h.

6.189.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::rhs  [private]
```

Definition at line 295 of file divides.h.

6.189.5.3 rhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::rhs_na [private]
```

Definition at line 296 of file divides.h.

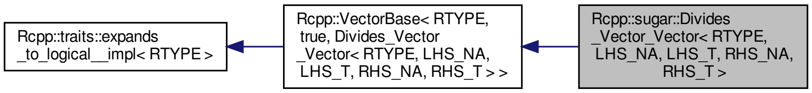
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

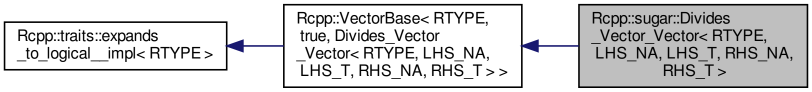
6.190 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, 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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >

Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.190.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file divides.h.

6.190.2 Member Typedef Documentation

6.190.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 34 of file divides.h.

6.190.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 31 of file divides.h.

6.190.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 35 of file divides.h.

6.190.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 32 of file divides.h.

6.190.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 33 of file divides.h.

6.190.3 Constructor & Destructor Documentation

6.190.3.1 Divides_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )   [inline]
```

Definition at line 37 of file divides.h.

6.190.4 Member Function Documentation

6.190.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 40 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.190.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 47 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.190.5 Member Data Documentation

6.190.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 50 of file divides.h.

Referenced by Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size().

6.190.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 51 of file divides.h.

Referenced by Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), and Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ]().

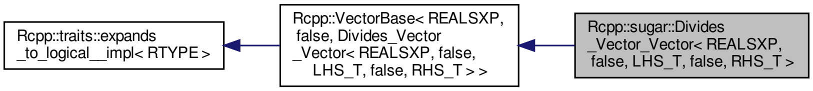
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

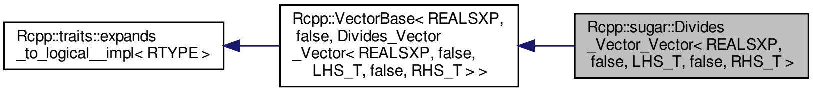
6.191 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Divides_Vector
_Vector< REALSXP, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Divides_Vector
_Vector< REALSXP, false, LHS_T, false, RHS_T > >
Rcpp::sugar::Divides
_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.191.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Definition at line 200 of file divides.h.

6.191.2 Member Typedef Documentation

6.191.2.1 LHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 205 of file divides.h.

6.191.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 203 of file divides.h.

6.191.2.3 RHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 206 of file divides.h.

6.191.2.4 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 204 of file divides.h.

6.191.3 Constructor & Destructor Documentation

6.191.3.1 Divides_Vector_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Divides_Vector_Vector
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 208 of file divides.h.

6.191.4 Member Function Documentation

6.191.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 211 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.191.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( )
const  [inline]
```

Definition at line 215 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.191.5 Member Data Documentation

6.191.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 218 of file divides.h.

6.191.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 219 of file divides.h.

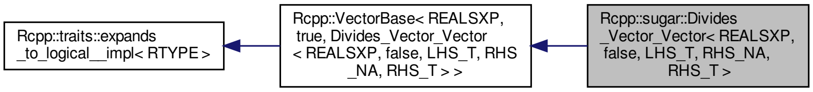
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

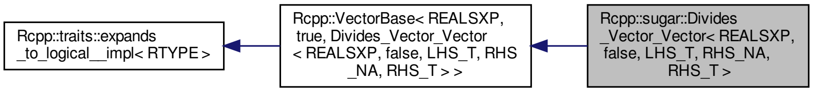
6.192 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector < REALSXP, false, LHS_T, RHS _NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector < REALSXP, false, LHS_T, RHS _NA, RHS_T > >
Rcpp::sugar::Divides _Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.192.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Definition at line 104 of file divides.h.

6.192.2 Member Typedef Documentation

6.192.2.1 LHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 109 of file divides.h.

6.192.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 107 of file divides.h.

6.192.2.3 RHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 110 of file divides.h.

6.192.2.4 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 108 of file divides.h.

6.192.3 Constructor & Destructor Documentation

6.192.3.1 Divides_Vector_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 112 of file divides.h.

6.192.4 Member Function Documentation

6.192.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 115 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.192.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 119 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.192.5 Member Data Documentation

6.192.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 122 of file divides.h.

6.192.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 123 of file divides.h.

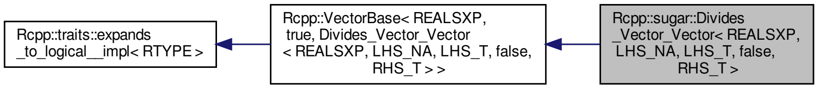
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

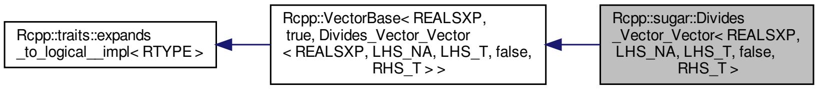
6.193 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.193.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Definition at line 153 of file divides.h.

6.193.2 Member Typedef Documentation

6.193.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 158 of file divides.h.

6.193.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS←
_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 156 of file divides.h.

6.193.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 159 of file divides.h.

6.193.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_←
NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 157 of file divides.h.

6.193.3 Constructor & Destructor Documentation

6.193.3.1 Divides_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Divides_Vector_Vector
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 161 of file divides.h.

6.193.4 Member Function Documentation

6.193.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 164 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.193.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( )
const  [inline]
```

Definition at line 167 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.193.5 Member Data Documentation

6.193.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 170 of file divides.h.

6.193.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 171 of file divides.h.

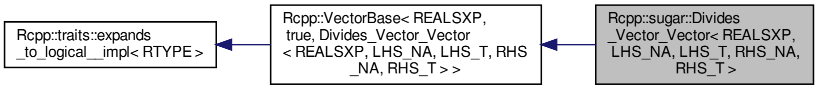
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

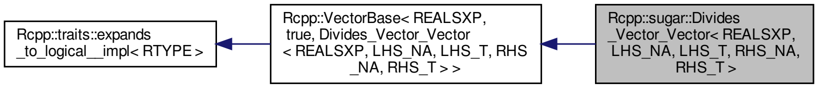
6.194 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS _NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS _NA, RHS_T > >
Rcpp::sugar::Divides _Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.194.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 55 of file divides.h.

6.194.2 Member Typedef Documentation

6.194.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 60 of file divides.h.

6.194.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS↩
_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 58 of file divides.h.

6.194.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 61 of file divides.h.

6.194.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS↩
_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 59 of file divides.h.

6.194.3 Constructor & Destructor Documentation

6.194.3.1 Divides_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 63 of file divides.h.

6.194.4 Member Function Documentation

6.194.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 66 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp↩ ::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.194.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 70 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.194.5 Member Data Documentation

6.194.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 73 of file divides.h.

6.194.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 74 of file divides.h.

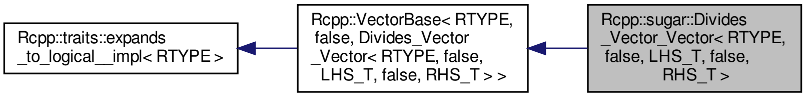
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

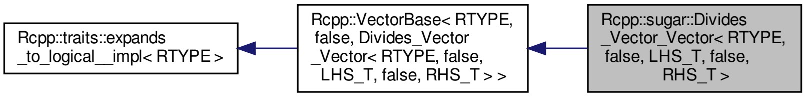
6.195 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Divides_Vector _Vector< RTYPE, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Divides_Vector _Vector< RTYPE, false, LHS_T, false, RHS_T > >
Rcpp::sugar::Divides _Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.195.1 Detailed Description

template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Definition at line 176 of file divides.h.

6.195.2 Member Typedef Documentation

6.195.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 182 of file divides.h.

6.195.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 179 of file divides.h.

6.195.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 183 of file divides.h.

6.195.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 180 of file divides.h.

6.195.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS↩
_T, false, RHS_T >::STORAGE
```

Definition at line 181 of file divides.h.

6.195.3 Constructor & Destructor Documentation

6.195.3.1 Divides_Vector_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Divides_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 185 of file divides.h.

6.195.4 Member Function Documentation

6.195.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 188 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.195.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 192 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.195.5 Member Data Documentation

6.195.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 195 of file divides.h.

6.195.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 196 of file divides.h.

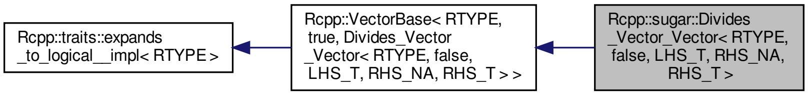
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

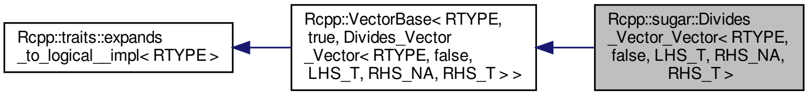
6.196 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.196.1 Detailed Description

template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Definition at line 79 of file divides.h.

6.196.2 Member Typedef Documentation

6.196.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 84 of file divides.h.

6.196.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 81 of file divides.h.

6.196.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 85 of file divides.h.

6.196.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 82 of file divides.h.

6.196.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS↩
_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 83 of file divides.h.

6.196.3 Constructor & Destructor Documentation

6.196.3.1 Divides_Vector_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 87 of file divides.h.

6.196.4 Member Function Documentation

6.196.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 90 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp↩ ::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.196.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 96 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.196.5 Member Data Documentation

6.196.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 99 of file divides.h.

6.196.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 100 of file divides.h.

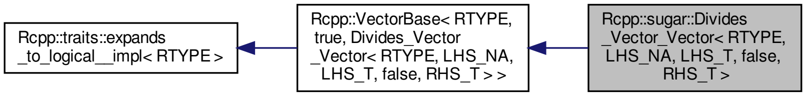
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

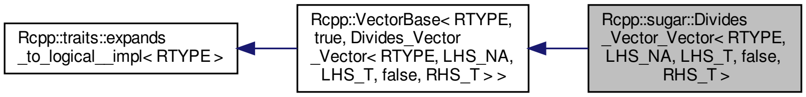
6.197 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Divides_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.197.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Definition at line 128 of file divides.h.

6.197.2 Member Typedef Documentation

6.197.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 134 of file divides.h.

6.197.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 131 of file divides.h.

6.197.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 135 of file divides.h.

6.197.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 132 of file divides.h.

6.197.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::STORAGE
```

Definition at line 133 of file divides.h.

6.197.3 Constructor & Destructor Documentation

6.197.3.1 Divides_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Divides_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 137 of file divides.h.

6.197.4 Member Function Documentation

6.197.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 140 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.197.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 145 of file divides.h.

References Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.197.5 Member Data Documentation

6.197.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 148 of file divides.h.

6.197.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 149 of file divides.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

6.198 Rcpp::DottedPairImpl< CLASS > Class Template Reference

```
#include <DottedPairImpl.h>
```

Public Member Functions

- template<typename T >
  void push_back (const T &object)
- template<typename T >
  void push_front (const T &object)
- template<typename T >
  void insert (const size_t &index, const T &object)
- template<typename T >
  void replace (const int &index, const T &object)
- R_xlen_t length () const
- R_xlen_t size () const
- void remove (const size_t &index)

Friends

- template<typename T >
  DottedPairImpl & operator<< (DottedPairImpl &os, const T &t)
- template<typename T >
  DottedPairImpl & operator>> (const T &t, DottedPairImpl &s)

6.198.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairImpl< CLASS >

Definition at line 24 of file DottedPairImpl.h.

6.198.2 Member Function Documentation

6.198.2.1 insert()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::insert (
             const size_t & index,
             const T & object )
```

insert an object at the given position, pushing other objects to the tail of the list

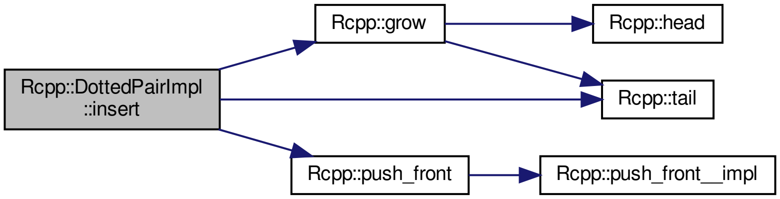
Parameters

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:



6.198.2.2 length()

```
template<typename CLASS >
R_xlen_t Rcpp::DottedPairImpl< CLASS >::length ( ) const  [inline]
```

Definition at line 66 of file DottedPairImpl.h.

6.198.2.3 push_back()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::push_back (
             const T & object )
```

wraps an object and add it at the end of the pairlist (this require traversing the entire pairlist)

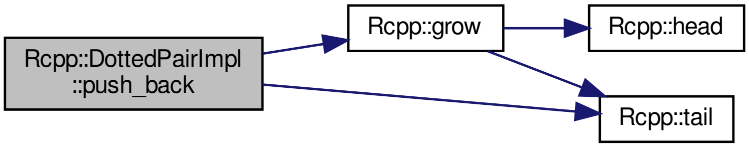
Parameters

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:



6.198.2.4 push_front()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::push_front (
            const T & object )
```

wraps an object and add it in front of the pairlist.

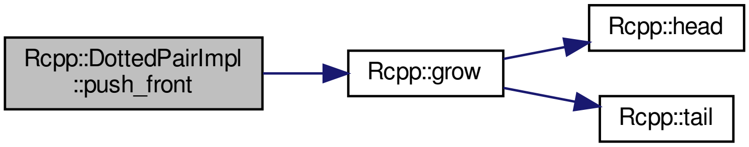
Parameters

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:



6.198.2.5 remove()

```
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 101 of file DottedPairImpl.h.

6.198.2.6 replace()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::replace (
             const int & index,
             const T & object )
```

replaces an element of the list

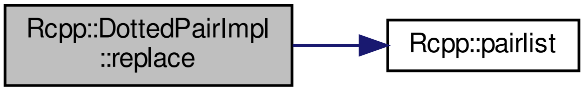
Parameters

index	position
object	object that can be wrapped

Definition at line 80 of file DottedPairImpl.h.

References Rcpp::pairlist().

Here is the call graph for this function:



6.198.2.7 size()

```
template<typename CLASS >
R_xlen_t Rcpp::DottedPairImpl< CLASS >::size ( ) const  [inline]
```

Definition at line 70 of file DottedPairImpl.h.

6.198.3 Friends And Related Function Documentation

6.198.3.1 operator<<

```
template<typename CLASS >
template<typename T >
DottedPairImpl& operator<< (
             DottedPairImpl< CLASS > & os,
             const T & t )  [friend]
```

Definition at line 82 of file DottedPairImpl.h.

6.198.3.2 operator>>

```
template<typename CLASS >
template<typename T >
DottedPairImpl& operator>> (
             const T & t,
             DottedPairImpl< CLASS > & s )  [friend]
```

Definition at line 88 of file DottedPairImpl.h.

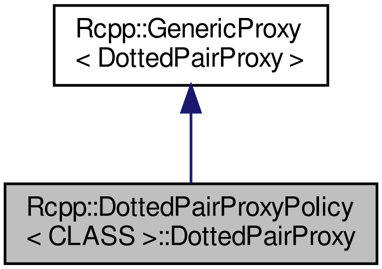
The documentation for this class was generated from the following file:

- inst/include/Rcpp/DottedPairImpl.h

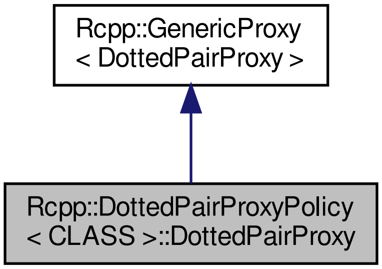
6.199 Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy Class Reference

```
#include <DottedPairProxy.h>
```

Inheritance diagram for Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy:



Collaboration diagram for Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy:



Public Member Functions

- DottedPairProxy (CLASS &v, int index_)
- DottedPairProxy & operator= (const DottedPairProxy &rhs)
- DottedPairProxy & operator= (SEXP rhs)
- template< typename T >
  DottedPairProxy & operator= (const T &rhs)

- template<typename T >
  DottedPairProxy & operator= (const traits::named_object< T > &rhs)
- template<typename T >
  operator T () const
- SEXP get () const
- operator SEXP () const
- DottedPairProxy & set (SEXP x)
- DottedPairProxy & set (SEXP x, const char ∗name)
- template<typename T >
  DottedPairProxyPolicy< CLASS >::DottedPairProxy & operator= (const T &rhs)
- template<typename T >
  DottedPairProxyPolicy< CLASS >::DottedPairProxy & operator= (const traits::named_object< T > &rhs)

Private Attributes

- SEXP node

6.199.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy

Definition at line 27 of file DottedPairProxy.h.

6.199.2 Constructor & Destructor Documentation

6.199.2.1 DottedPairProxy()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy (
             CLASS & v,
             int index_ )  [inline]
```

Definition at line 29 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.199.3 Member Function Documentation

6.199.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get ( ) const  [inline]
```

Definition at line 56 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator SEXP(), and Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

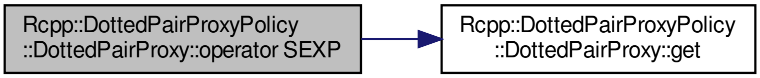
6.199.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator SEXP ( ) const  [inline]
```

Definition at line 59 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get().

Here is the call graph for this function:


Rcpp::DottedPairProxyPolicy
::DottedPairProxy::operator SEXP
Rcpp::DottedPairProxyPolicy
::DottedPairProxy::get

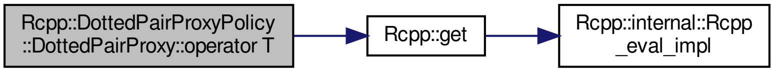
6.199.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator T
```

Definition at line 165 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



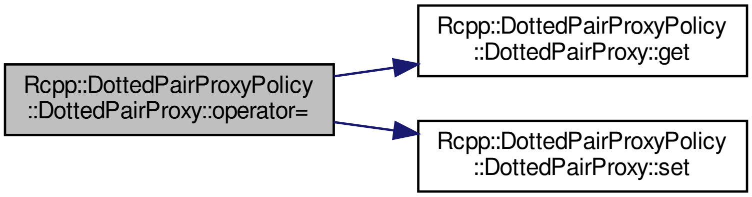
6.199.3.4 operator=() [1/6]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
             const DottedPairProxy & rhs )  [inline]
```

Definition at line 41 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get(), and Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set().

Here is the call graph for this function:


Rcpp::DottedPairProxyPolicy ::DottedPairProxy::operator=
Rcpp::DottedPairProxyPolicy ::DottedPairProxy::get
Rcpp::DottedPairProxyPolicy ::DottedPairProxy::set

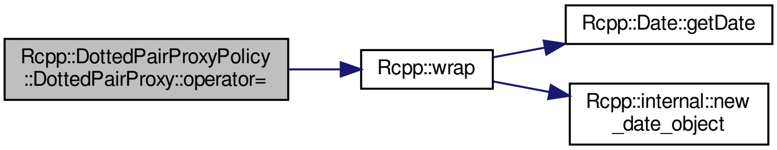
6.199.3.5 operator=() [2/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxyPolicy<CLASS>::DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::Dotted←
PairProxy::operator= (
             const T & rhs )
```

Definition at line 151 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::DottedPairProxyPolicy ::DottedPairProxy::operator=
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.199.3.6 operator=() [3/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
            const T & rhs )
```

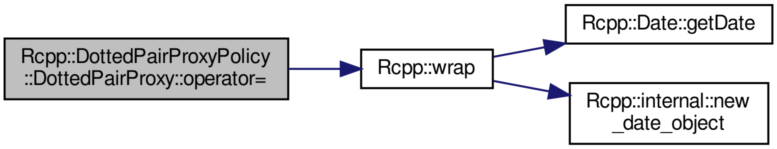
6.199.3.7 operator=() [4/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxyPolicy<CLASS>::DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::Dotted↩
PairProxy::operator= (
            const traits::named_object< T > & rhs )
```

Definition at line 159 of file proxy.h.

References Rcpp::traits::named_object< T >::name, Rcpp::traits::named_object< T >::object, and Rcpp::wrap().

Here is the call graph for this function:



6.199.3.8 operator=() [5/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
            const traits::named_object< T > & rhs )
```

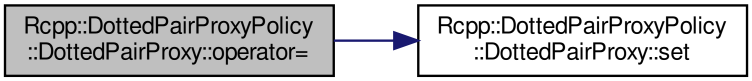
6.199.3.9 operator=() [6/6]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
             SEXP rhs )  [inline]
```

Definition at line 44 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set().

Here is the call graph for this function:


Rcpp::DottedPairProxyPolicy ::DottedPairProxy::operator=
Rcpp::DottedPairProxyPolicy ::DottedPairProxy::set

6.199.3.10 set() [1/2]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (
             SEXP x )  [inline]
```

Definition at line 62 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

6.199.3.11 set() [2/2]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (
             SEXP x,
             const char * name )  [inline]
```

Definition at line 66 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

6.199.4 Member Data Documentation

6.199.4.1 node

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node  [private]
```

Definition at line 74 of file DottedPairProxy.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy(), Rcpp::DottedPair↩ProxyPolicy< CLASS >::DottedPairProxy::get(), and Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/DottedPairProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.200 Rcpp::DottedPairProxyPolicy< CLASS > Class Template Reference

```
#include <DottedPairProxy.h>
```

Classes

- class const_DottedPairProxy
- class DottedPairProxy

Public Member Functions

- DottedPairProxy operator[ ] (int i)
- const_DottedPairProxy operator[ ] (int i) const

6.200.1 Detailed Description

template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >

Definition at line 24 of file DottedPairProxy.h.

6.200.2 Member Function Documentation

6.200.2.1 operator[]() [1/2]

```
template<typename CLASS >
DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[] (
             int i )  [inline]
```

Definition at line 105 of file DottedPairProxy.h.

6.200.2.2 operator[]() [2/2]

```
template<typename CLASS >
const_DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[] (
             int i ) const  [inline]
```

Definition at line 108 of file DottedPairProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/DottedPairProxy.h

6.201 Rcpp::internal::element_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

Public Types

- typedef ::Rcpp::traits::storage_type< RTYPE >::type target

Static Public Member Functions

- template<typename T >
  static target get (const T &input)
- static target get (const target &input)

6.201.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::element_converter< RTYPE >

Definition at line 28 of file converter.h.

6.201.2 Member Typedef Documentation

6.201.2.1 target

```
template<int RTYPE>
typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::element_converter< RTYPE >↩
::target
```

Definition at line 30 of file converter.h.

6.201.3 Member Function Documentation

6.201.3.1 get() [1/2]

```
template<int RTYPE>
template<typename T >
static target Rcpp::internal::element_converter< RTYPE >::get (
             const T & input )  [inline], [static]
```

Definition at line 33 of file converter.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill↩_diag__dispatch(), Rcpp::Vector< RTYPE, StoragePolicy >::insert(), Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()(), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >↩::operator[ ](), Rcpp::Vector< RTYPE, StoragePolicy >::push_back(), Rcpp::Vector< RTYPE, StoragePolicy >::push↩_front(), Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch(), and Rcpp::Vector< RTYPE, Storage↩Policy >::replace_element__dispatch__isArgument().

6.201.3.2 get() [2/2]

```
template<int RTYPE>
static target Rcpp::internal::element_converter< RTYPE >::get (
            const target & input ) [inline], [static]
```

Definition at line 37 of file converter.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.202 Rcpp::traits::enable_if< B, T > Struct Template Reference

```
#include <enable_if.h>
```

6.202.1 Detailed Description

template<bool B, typename T = void>
struct Rcpp::traits::enable_if< B, T >

Definition at line 8 of file enable_if.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/enable_if.h

6.203 Rcpp::traits::enable_if< true, T > Struct Template Reference

```
#include <enable_if.h>
```

Public Types

- typedef T type

6.203.1 Detailed Description

template<typename T>
struct Rcpp::traits::enable_if< true, T >

Definition at line 11 of file enable_if.h.

6.203.2 Member Typedef Documentation

6.203.2.1 type

```
template<typename T >
typedef T Rcpp::traits::enable_if< true, T >::type
```

Definition at line 12 of file enable_if.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/enable_if.h

6.204 Rcpp::enum_< Enum, Parent > Class Template Reference

```
#include <Module.h>
```

Public Types

- typedef enum_< Enum, Parent > self

Public Member Functions

- enum_ (const char ∗name_)
- ∼enum_ ()
- self & value (const char ∗name_, Enum value_)

Private Types

- typedef std::map< std::string, int > MAP
- typedef MAP::value_type PAIR

Private Attributes

- std::string name
- MAP values
- std::string parent_typeinfo_name

6.204.1 Detailed Description

template<typename Enum, typename Parent>
class Rcpp::enum_< Enum, Parent >

Definition at line 343 of file Module.h.

6.204.2 Member Typedef Documentation

6.204.2.1 MAP

```
template<typename Enum , typename Parent >
typedef std::map< std::string, int > Rcpp::enum_< Enum, Parent >::MAP  [private]
```

Definition at line 363 of file Module.h.

6.204.2.2 PAIR

```
template<typename Enum , typename Parent >
typedef MAP::value_type Rcpp::enum_< Enum, Parent >::PAIR  [private]
```

Definition at line 364 of file Module.h.

6.204.2.3 self

```
template<typename Enum , typename Parent >
typedef enum_<Enum,Parent> Rcpp::enum_< Enum, Parent >::self
```

Definition at line 345 of file Module.h.

6.204.3 Constructor & Destructor Documentation

6.204.3.1 enum_()

```
template<typename Enum , typename Parent >
Rcpp::enum_< Enum, Parent >::enum_ (
             const char * name_ )  [inline]
```

Definition at line 347 of file Module.h.

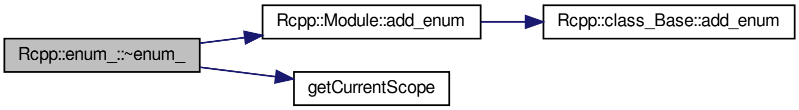
6.204.3.2 ∼enum_()

```
template<typename Enum , typename Parent >
Rcpp::enum_< Enum, Parent >::∼enum_ ( )  [inline]
```

Definition at line 350 of file Module.h.

References Rcpp::Module::add_enum(), getCurrentScope(), Rcpp::enum_< Enum, Parent >::name, Rcpp::enum_< Enum, Parent >::parent_typeinfo_name, and Rcpp::enum_< Enum, Parent >::values.

Here is the call graph for this function:



6.204.4 Member Function Documentation

6.204.4.1 value()

```
template<typename Enum , typename Parent >
self& Rcpp::enum_< Enum, Parent >::value (
             const char * name_,
             Enum value_ )  [inline]
```

Definition at line 355 of file Module.h.

References Rcpp::enum_< Enum, Parent >::values.

6.204.5 Member Data Documentation

6.204.5.1 name

```
template<typename Enum , typename Parent >
std::string Rcpp::enum_< Enum, Parent >::name  [private]
```

Definition at line 362 of file Module.h.

Referenced by Rcpp::enum_< Enum, Parent >::∼enum_().

6.204.5.2 parent_typeinfo_name

```
template<typename Enum , typename Parent >
std::string Rcpp::enum_< Enum, Parent >::parent_typeinfo_name  [private]
```

Definition at line 366 of file Module.h.

Referenced by Rcpp::enum_< Enum, Parent >::∼enum_().

6.204.5.3 values

```
template<typename Enum , typename Parent >
MAP Rcpp::enum_< Enum, Parent >::values  [private]
```

Definition at line 365 of file Module.h.

Referenced by Rcpp::enum_< Enum, Parent >::value(), and Rcpp::enum_< Enum, Parent >::∼enum_().

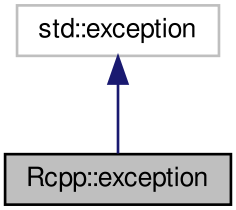
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

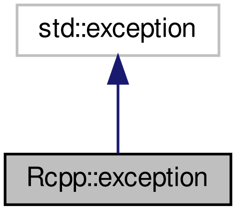
6.205 Rcpp::exception Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::exception:



Collaboration diagram for Rcpp::exception:



Public Member Functions

- exception (const char ∗message_, bool include_call=RCPP_DEFAULT_INCLUDE_CALL)
- exception (const char ∗message_, const char ∗, int, bool include_call=RCPP_DEFAULT_INCLUDE_CALL)
- bool include_call () const
- virtual ∼exception () throw ()
- virtual const char ∗ what () const throw ()
- void copy_stack_trace_to_r () const

Private Member Functions

- void record_stack_trace ()

Private Attributes

- std::string message
- bool include_call_
- std::vector< std::string > stack

6.205.1 Detailed Description

Definition at line 36 of file exceptions.h.

6.205.2 Constructor & Destructor Documentation

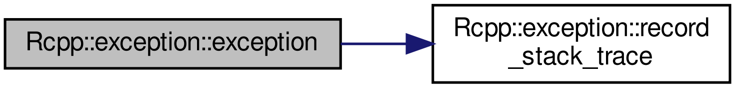
6.205.2.1 exception() [1/2]

```
Rcpp::exception::exception (
             const char * message_,
             bool include_call = RCPP_DEFAULT_INCLUDE_CALL )  [inline], [explicit]
```

Definition at line 38 of file exceptions.h.

References record_stack_trace().

Here is the call graph for this function:



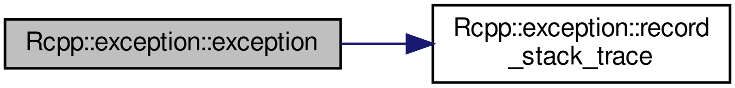
6.205.2.2 exception() [2/2]

```
Rcpp::exception::exception (
            const char * message_,
            const char * ,
            int ,
            bool include_call = RCPP_DEFAULT_INCLUDE_CALL )  [inline]
```

Definition at line 43 of file exceptions.h.

References record_stack_trace().

Here is the call graph for this function:



6.205.2.3 ∼exception()

```
virtual Rcpp::exception::∼exception ( ) throw ( )   [inline], [virtual]
```

Definition at line 51 of file exceptions.h.

6.205.3 Member Function Documentation

6.205.3.1 copy_stack_trace_to_r()

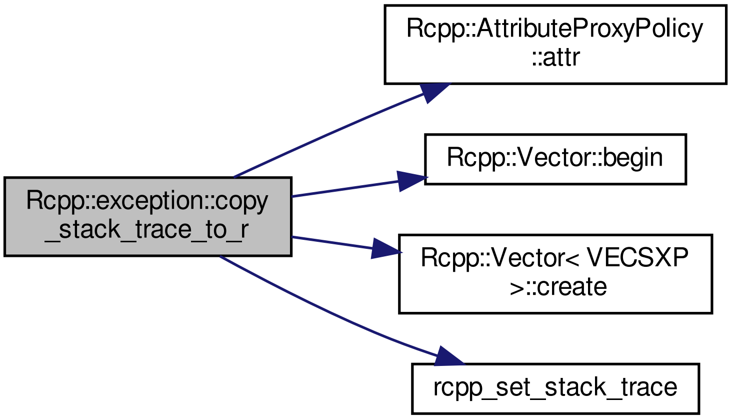
```
void Rcpp::exception::copy_stack_trace_to_r ( ) const  [inline]
```

Definition at line 92 of file exceptions_impl.h.

References Rcpp::_, Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< VECSXP >::create(), rcpp_set_stack_trace(), and stack.

Referenced by rcpp_exception_to_r_condition().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::exception::copy _stack_trace_to_r
Rcpp::Vector< VECSXP >::create
rcpp_set_stack_trace

6.205.3.2 include_call()

```
bool Rcpp::exception::include_call ( ) const  [inline]
```

Definition at line 48 of file exceptions.h.

References include_call_.

Referenced by rcpp_exception_to_r_condition().

6.205.3.3 record_stack_trace()

```
void Rcpp::exception::record_stack_trace ( )  [inline], [private]
```

Definition at line 74 of file exceptions_impl.h.

References stack.

Referenced by exception().

6.205.3.4 what()

```
virtual const char* Rcpp::exception::what ( ) const throw ( )   [inline], [virtual]
```

Definition at line 52 of file exceptions.h.

References message.

6.205.4 Member Data Documentation

6.205.4.1 include_call_

```
bool Rcpp::exception::include_call_  [private]
```

Definition at line 58 of file exceptions.h.

Referenced by include_call().

6.205.4.2 message

```
std::string Rcpp::exception::message  [private]
```

Definition at line 57 of file exceptions.h.

Referenced by what().

6.205.4.3 stack

```
std::vector<std::string> Rcpp::exception::stack [private]
```

Definition at line 59 of file exceptions.h.

Referenced by copy_stack_trace_to_r(), and record_stack_trace().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/exceptions.h
- inst/include/Rcpp/exceptions_impl.h

6.206 Rcpp::algorithm::helpers::exp Struct Reference

```
#include <algorithm.h>
```

Public Member Functions

- template<typename T >
  double operator() (T val)

6.206.1 Detailed Description

Definition at line 206 of file algorithm.h.

6.206.2 Member Function Documentation

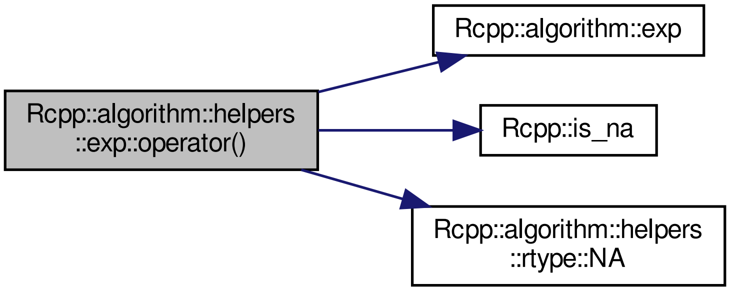
6.206.2.1 operator()()

```
template<typename T >
double Rcpp::algorithm::helpers::exp::operator() (
            T val )  [inline]
```

Definition at line 208 of file algorithm.h.

References Rcpp::algorithm::exp(), Rcpp::is_na(), and Rcpp::algorithm::helpers::rtype< T >::NA().

Here is the call graph for this function:


Rcpp::algorithm::helpers ::exp::operator()
Rcpp::algorithm::exp
Rcpp::is_na
Rcpp::algorithm::helpers ::rtype::NA

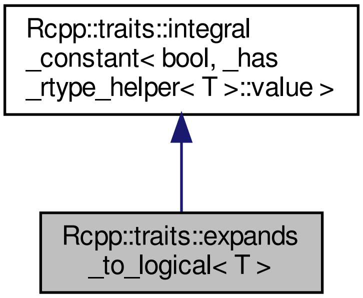
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

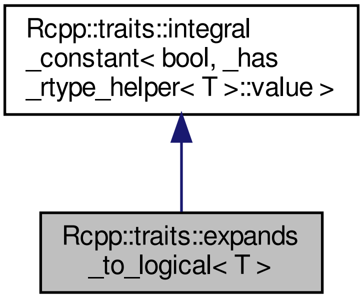
6.207 Rcpp::traits::expands_to_logical< T > Struct Template Reference

```
#include <expands_to_logical.h>
```

Inheritance diagram for Rcpp::traits::expands_to_logical< T >:



Collaboration diagram for Rcpp::traits::expands_to_logical< T >:



Additional Inherited Members

6.207.1 Detailed Description

template<typename T>
struct Rcpp::traits::expands_to_logical< T >

Definition at line 54 of file expands_to_logical.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

6.208 Rcpp::traits::expands_to_logical__impl< RTYPE > Struct Template Reference

```
#include <expands_to_logical.h>
```

Inherited by Rcpp::MatrixBase< LGLSXP, false, UpperTri< RTYPE, NA, T > >, Rcpp::MatrixBase< RTYPE, true, SubMatrix< RTYPE >
Rcpp::MatrixBase< INTSXP, false, Row< RTYPE, LHS_NA, LHS_T > >, Rcpp::MatrixBase< RTYPE, NA, Diag_Maker< RTYPE, NA,
Rcpp::MatrixBase< INTSXP, false, Col< RTYPE, LHS_NA, LHS_T > >, Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBind
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >, Rcpp::MatrixBase< RTYPE,
Rcpp::MatrixBase< LGLSXP, false, LowerTri< RTYPE, NA, T > >, Rcpp::MatrixBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::re
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >, Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage >
Rcpp::MatrixBase< RTYPE, NA, T >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive< RTYPE, NA, T > >,
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE, tru
Rcpp::VectorBase< RTYPE,(COND_NA||LHS_NA||RHS_NA), IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP, NA
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive
Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, true, Pmin_Vector_Primitive< RTYPE, LHS_N
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, false, N
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTY
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_1< NA, RESULT_TYPE, U1, T1 > >,
Rcpp::VectorBase< RTYPE, NA, Rep_each< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), Pmax_Vector_Vec
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, true, Divides_Primitive_Vector< REALSXP, false, T > >, Rcpp::VectorBase< REALSXP, true, Divides_P
Rcpp::VectorBase< REALSXP, NA, P3< RTYPE, NA, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< T >::rtype, true, Rep
Rcpp::VectorBase< REALSXP, NA, Pow< RTYPE, NA, T, EXPONENT_TYPE > >, Rcpp::VectorBase< RTYPE, true, Pmax_Vector_Pri
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > >,
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, t
Rcpp::VectorBase< REALSXP, false, Vectorized_INTSXP< Func, false, VEC > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vec
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< LGLSXP, true, And_LogicalExpre
Rcpp::VectorBase< INTSXP, false, Range >, Rcpp::VectorBase< RTYPE, true, ConstMatrixRow< RTYPE > >,
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, NA, Rep_len< RTYPE,
Rcpp::VectorBase< REALSXP, false, Pow< INTSXP, false, T, EXPONENT_TYPE > >, Rcpp::VectorBase< REALSXP, true, Plus_Vector
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, true, Minus_Primitive_V
Rcpp::VectorBase< LGLSXP, false, IsNaN< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector<
Rcpp::VectorBase< RTYPE, NA, Diag_Extractor< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, RangeIndexer< RTYPE, NA,
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REALSXP, f
Rcpp::VectorBase< RTYPE, NA, Head< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSX
Rcpp::VectorBase< REALSXP, NA, Vectorized_INTSXP< Func, NA, VEC > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RE
Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Rev< RTYPE, NA, T > >,
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive_nona< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Plus_Vector_
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, Divides_Primitive_
Rcpp::VectorBase< RTYPE, true, Divides_Primitive_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, false, Diff< RTYPE, fals
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE
Rcpp::VectorBase< LGLSXP, false, Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > >,
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_tra
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Mapply_2< RTYPE, NA_
Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector<
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >,

Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sa
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sa
Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< LGLSXP, fals
Rcpp::VectorBase< RTYPE, true, ConstMatrixColumn< RTYPE > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TY
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP,
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive_nona< REALSXP, NA, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, false, VEC_TYPE > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive
Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, NA, Clamp_Primitive_Vector_
Rcpp::VectorBase< RTYPE, true, MatrixColumn< RTYPE > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive_nona< RTYPE, false, T > >, Rcpp::VectorBase< REALSXP, true, Times_Ve
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive_nona< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Pow< INTSX
Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > >, Rcpp::VectorBase< LGLSXP, true, Comparator_With_
Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D0< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, NA, Vectorized< Func, NA, VEC > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REA
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive_nona< REALSXP, false, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexpty
Rcpp::VectorBase< INTSXP, false, SeqLen >, Rcpp::VectorBase< RTYPE, false, Minus_Vector_Vector< RTYPE, false, LHS_T, false, R
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYP
Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REA
Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REAL
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Q2< RTYPE, NA,
Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, fa
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< Rcpp:
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_trai
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsInfinite< RTYPE, NA, V
Rcpp::VectorBase< LGLSXP, false, IsFinite< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< RTYPE, false, Divides_Vector_Vector<
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< LGLSXP,
Rcpp::VectorBase< REALSXP, NA, Q1< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, NA, D1< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Tail< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE,
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > >, Rcpp::VectorBase< RTYPE, true, Comparator_V
Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > >,
Rcpp::VectorBase< unary_minus_result_type< RTYPE >::value, NA, UnaryMinus_Vector< unary_minus_result_type< RTYPE >::valu
Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< Rcpp::traits::r_
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >,
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, t
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Pr
Rcpp::VectorBase< REALSXP, false, Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSX
Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D3< RTYPE, NA, T > >,
Rcpp::MatrixBase< RTYPE, na, MATRIX >, and Rcpp::VectorBase< RTYPE, na, VECTOR >.

6.208.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::expands_to_logical__impl< RTYPE >

Definition at line 33 of file expands_to_logical.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

6.209 Rcpp::traits::expands_to_logical__impl< LGLSXP > Struct Reference

```
#include <expands_to_logical.h>
```

Classes

- struct r_expands_to_logical

6.209.1 Detailed Description

Definition at line 36 of file expands_to_logical.h.

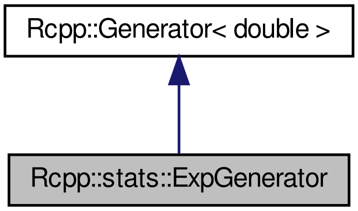
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

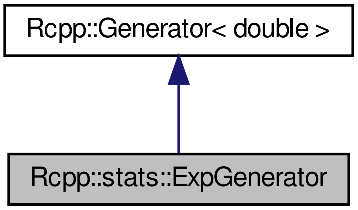
6.210 Rcpp::stats::ExpGenerator Class Reference

```
#include <rexp.h>
```

Inheritance diagram for Rcpp::stats::ExpGenerator:



Collaboration diagram for Rcpp::stats::ExpGenerator:



Public Member Functions

- ExpGenerator (double scale_)
- double operator() () const

Private Attributes

- double scale

Additional Inherited Members

6.210.1 Detailed Description

Definition at line 28 of file rexp.h.

6.210.2 Constructor & Destructor Documentation

6.210.2.1 ExpGenerator()

```
Rcpp::stats::ExpGenerator::ExpGenerator (
            double scale_ )  [inline]
```

Definition at line 31 of file rexp.h.

6.210.3 Member Function Documentation

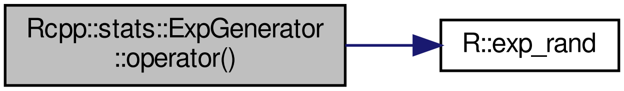
6.210.3.1 operator()()

```
double Rcpp::stats::ExpGenerator::operator() ( ) const  [inline]
```

Definition at line 33 of file rexp.h.

References R::exp_rand(), and scale.

Here is the call graph for this function:



6.210.4 Member Data Documentation

6.210.4.1 scale

```
double Rcpp::stats::ExpGenerator::scale  [private]
```

Definition at line 38 of file rexp.h.

Referenced by operator()().

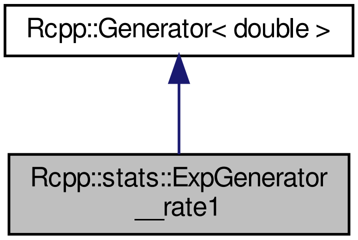
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rexp.h

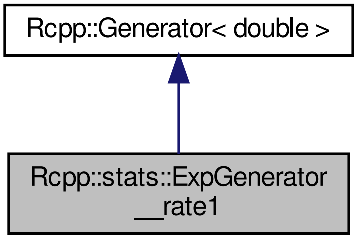
6.211 Rcpp::stats::ExpGenerator__rate1 Class Reference

```
#include <rexp.h>
```

Inheritance diagram for Rcpp::stats::ExpGenerator__rate1:



Collaboration diagram for Rcpp::stats::ExpGenerator__rate1:



Public Member Functions

- ExpGenerator__rate1 ()
- double operator() () const

Additional Inherited Members

6.211.1 Detailed Description

Definition at line 41 of file rexp.h.

6.211.2 Constructor & Destructor Documentation

6.211.2.1 ExpGenerator__rate1()

```
Rcpp::stats::ExpGenerator__rate1::ExpGenerator__rate1 ( )  [inline]
```

Definition at line 43 of file rexp.h.

6.211.3 Member Function Documentation

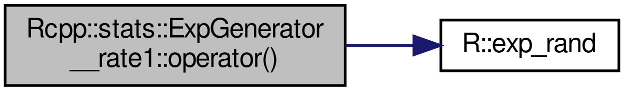
6.211.3.1 operator()()

```
double Rcpp::stats::ExpGenerator__rate1::operator() ( ) const  [inline]
```

Definition at line 44 of file rexp.h.

References R::exp_rand().

Here is the call graph for this function:

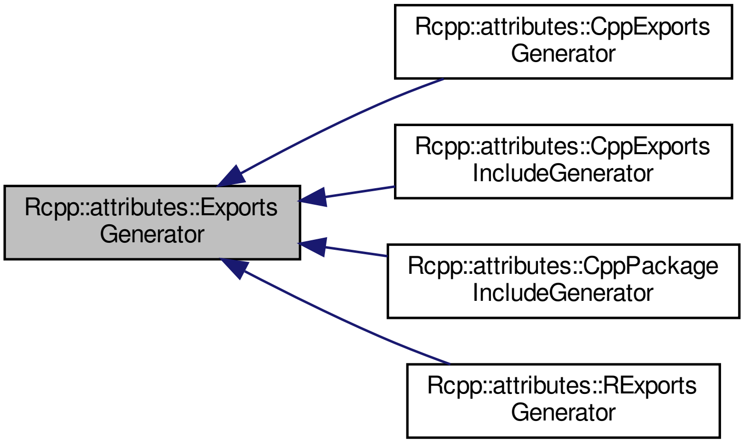


The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rexp.h

6.212 Rcpp::attributes::ExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::ExportsGenerator:


Rcpp::attributes::CppExports Generator
Rcpp::attributes::CppExports IncludeGenerator
Rcpp::attributes::Exports Generator
Rcpp::attributes::CppPackage IncludeGenerator
Rcpp::attributes::RExports Generator

Public Member Functions

- virtual ∼ExportsGenerator ()
- const std::string & targetFile () const
- const std::string & package () const
- const std::string & packageCpp () const
- const std::string packageCppPrefix () const
- virtual void writeBegin ()=0
- void writeFunctions (const SourceFileAttributes &attributes, bool verbose)
- virtual void writeEnd (bool hasPackageInit)=0
- virtual bool commit (const std::vector< std::string > &includes)=0
- bool remove ()
- operator std::ostream & ()

Protected Member Functions

- ExportsGenerator (const std::string &targetFile, const std::string &package, const std::string &commentPrefix)
- std::ostream & ostr ()
- bool hasCppInterface () const
- std::string exportValidationFunction ()
- std::string exportValidationFunctionRegisteredName ()
- std::string registerCCallableExportedName ()
- bool commit (const std::string &preamble=std::string())
- std::string dotNameHelper (const std::string &name) const

Private Member Functions

- ExportsGenerator (const ExportsGenerator &)
- ExportsGenerator & operator= (const ExportsGenerator &)
- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)=0
- bool isSafeToOverwrite () const
- std::string generatorToken () const

Private Attributes

- std::string targetFile_
- std::string package_
- std::string packageCpp_
- std::string commentPrefix_
- std::string existingCode_
- std::ostringstream codeStream_
- bool hasCppInterface_

6.212.1 Detailed Description

Definition at line 599 of file attributes.cpp.

6.212.2 Constructor & Destructor Documentation

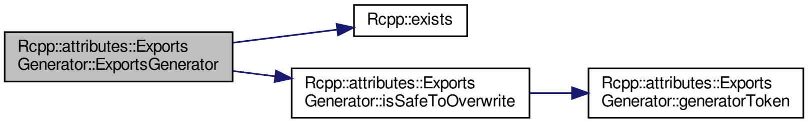
6.212.2.1 ExportsGenerator() [1/2]

```
Rcpp::attributes::ExportsGenerator::ExportsGenerator (
            const std::string & targetFile,
            const std::string & package,
            const std::string & commentPrefix )  [protected]
```

Definition at line 1866 of file attributes.cpp.

References existingCode_, Rcpp::exists(), isSafeToOverwrite(), packageCpp_, and targetFile_.

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::ExportsGenerator
Rcpp::exists
Rcpp::attributes::Exports
Generator::isSafeToOverwrite
Rcpp::attributes::Exports
Generator::generatorToken

6.212.2.2 ExportsGenerator() [2/2]

```
Rcpp::attributes::ExportsGenerator::ExportsGenerator (
             const ExportsGenerator &  )  [private]
```

6.212.2.3 ∼ExportsGenerator()

```
virtual Rcpp::attributes::ExportsGenerator::∼ExportsGenerator ( )  [inline], [virtual]
```

Definition at line 611 of file attributes.cpp.

6.212.3 Member Function Documentation

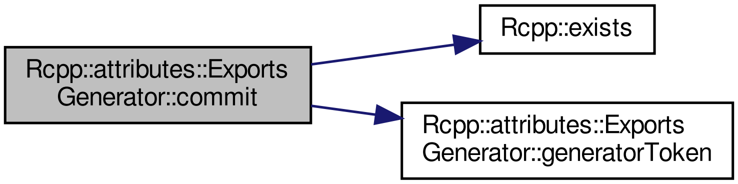
6.212.3.1 commit() [1/2]

```
bool Rcpp::attributes::ExportsGenerator::commit (
             const std::string & preamble = std::string() )  [protected]
```

Definition at line 1905 of file attributes.cpp.

References codeStream_, commentPrefix_, existingCode_, Rcpp::exists(), generatorToken(), and targetFile_.

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::commit
Rcpp::exists
Rcpp::attributes::Exports
Generator::generatorToken

6.212.3.2 commit() [2/2]

```
virtual bool Rcpp::attributes::ExportsGenerator::commit (
             const std::vector< std::string > & includes )  [pure virtual]
```

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncluc
and Rcpp::attributes::CppExportsGenerator.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppPackageIncludeGenerator::commit(), and Rcpp::attributes::RExportsGenerator::commit().

6.212.3.3 dotNameHelper()

```
std::string Rcpp::attributes::ExportsGenerator::dotNameHelper (
             const std::string & name ) const  [protected]
```

Definition at line 1950 of file attributes.cpp.

6.212.3.4 doWriteFunctions()

```
virtual void Rcpp::attributes::ExportsGenerator::doWriteFunctions (
             const SourceFileAttributes & attributes,
             bool verbose )  [private], [pure virtual]
```

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppExportsIncludeGenerator, Rcpp::attributes::CppExportsGener
and Rcpp::attributes::CppPackageIncludeGenerator.

Referenced by writeFunctions().

6.212.3.5 exportValidationFunction()

```
std::string Rcpp::attributes::ExportsGenerator::exportValidationFunction ( )  [inline], [protected]
```

Definition at line 647 of file attributes.cpp.

Referenced by exportValidationFunctionRegisteredName(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

6.212.3.6 exportValidationFunctionRegisteredName()

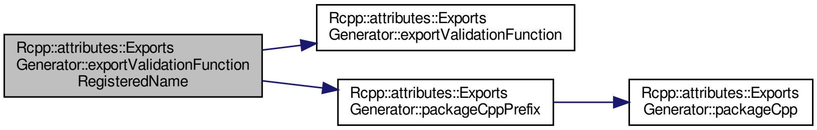
```
std::string Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName ( )  [inline],
[protected]
```

Definition at line 650 of file attributes.cpp.

References exportValidationFunction(), and packageCppPrefix().

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), and Rcpp::attributes::CppExports↵ Generator::writeEnd().

Here is the call graph for this function:


Rcpp::attributes::Exports Generator::exportValidationFunction RegisteredName
Rcpp::attributes::Exports Generator::exportValidationFunction
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp

6.212.3.7 generatorToken()

```
std::string Rcpp::attributes::ExportsGenerator::generatorToken ( ) const  [inline], [private]
```

Definition at line 681 of file attributes.cpp.

Referenced by commit(), and isSafeToOverwrite().

6.212.3.8 hasCppInterface()

```
bool Rcpp::attributes::ExportsGenerator::hasCppInterface ( ) const  [inline], [protected]
```

Definition at line 642 of file attributes.cpp.

References hasCppInterface_.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator↵ ::commit(), Rcpp::attributes::CppPackageIncludeGenerator::commit(), Rcpp::attributes::CppExportsGenerator::write↵ End(), Rcpp::attributes::CppPackageIncludeGenerator::writeEnd(), and Rcpp::attributes::RExportsGenerator::write↵ End().

6.212.3.9 isSafeToOverwrite()

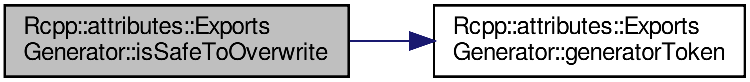
```
bool Rcpp::attributes::ExportsGenerator::isSafeToOverwrite ( ) const  [inline], [private]
```

Definition at line 674 of file attributes.cpp.

References existingCode_, and generatorToken().

Referenced by ExportsGenerator().

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::isSafeToOverwrite
Rcpp::attributes::Exports
Generator::generatorToken

6.212.3.10 operator std::ostream &()

```
Rcpp::attributes::ExportsGenerator::operator std::ostream & ( )  [inline]
```

Definition at line 631 of file attributes.cpp.

References codeStream_.

6.212.3.11 operator=()

```
ExportsGenerator& Rcpp::attributes::ExportsGenerator::operator= (
             const ExportsGenerator &  )  [private]
```

6.212.3.12 ostr()

```
std::ostream& Rcpp::attributes::ExportsGenerator::ostr ( )  [inline], [protected]
```

Definition at line 638 of file attributes.cpp.

References codeStream_.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator↩::commit(), Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsInclude↩Generator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::Cpp↩ExportsIncludeGenerator::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().

6.212.3.13 package()

```
const std::string& Rcpp::attributes::ExportsGenerator::package ( ) const  [inline]
```

Definition at line 615 of file attributes.cpp.

References package_.

Referenced by Rcpp::attributes::RExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsInclude↩Generator::getCCallable(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), Rcpp::attributes::CppExports↩IncludeGenerator::writeBegin(), and Rcpp::attributes::RExportsGenerator::writeEnd().

6.212.3.14 packageCpp()

```
const std::string& Rcpp::attributes::ExportsGenerator::packageCpp ( ) const  [inline]
```

Definition at line 616 of file attributes.cpp.

References packageCpp_.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppExportsInclude↩Generator::getHeaderGuard(), Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard(), packageCpp↩Prefix(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), Rcpp::attributes::CppExportsGenerator::write↩End(), and Rcpp::attributes::CppPackageIncludeGenerator::writeEnd().

6.212.3.15 packageCppPrefix()

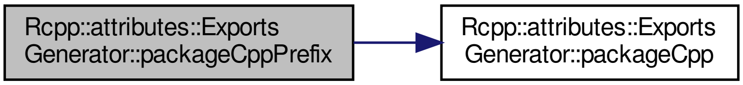
```
const std::string Rcpp::attributes::ExportsGenerator::packageCppPrefix ( ) const  [inline]
```

Definition at line 617 of file attributes.cpp.

References packageCpp().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), exportValidationFunctionRegisteredName(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), registerCCallableExportedName(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::packageCppPrefix
Rcpp::attributes::Exports
Generator::packageCpp

6.212.3.16 registerCCallableExportedName()

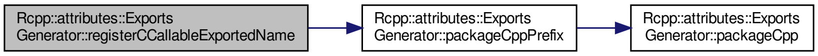
```
std::string Rcpp::attributes::ExportsGenerator::registerCCallableExportedName ( )  [inline], [protected]
```

Definition at line 653 of file attributes.cpp.

References packageCppPrefix().

Referenced by Rcpp::attributes::CppExportsGenerator::writeEnd(), and Rcpp::attributes::RExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::registerCCallableExportedName
Rcpp::attributes::Exports
Generator::packageCppPrefix
Rcpp::attributes::Exports
Generator::packageCpp

6.212.3.17 remove()

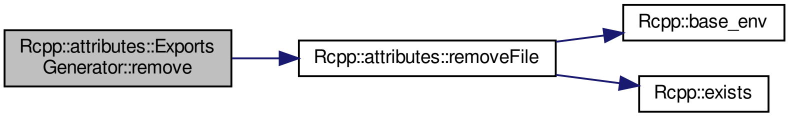
```
bool Rcpp::attributes::ExportsGenerator::remove ( )
```

Definition at line 1945 of file attributes.cpp.

References Rcpp::attributes::removeFile(), and targetFile_.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), and Rcpp::attributes::CppPackageIncludeGenerator::commit().

Here is the call graph for this function:



6.212.3.18 targetFile()

```
const std::string& Rcpp::attributes::ExportsGenerator::targetFile ( ) const [inline]
```

Definition at line 614 of file attributes.cpp.

References targetFile_.

Referenced by Rcpp::attributes::CppExportsGenerator::writeEnd().

6.212.3.19 writeBegin()

```
virtual void Rcpp::attributes::ExportsGenerator::writeBegin ( ) [pure virtual]
```

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsIncluc
and Rcpp::attributes::CppExportsGenerator.

6.212.3.20 writeEnd()

```
virtual void Rcpp::attributes::ExportsGenerator::writeEnd (
             bool hasPackageInit )  [pure virtual]
```

Implemented in Rcpp::attributes::RExportsGenerator, Rcpp::attributes::CppPackageIncludeGenerator, Rcpp::attributes::CppExportsInclud
and Rcpp::attributes::CppExportsGenerator.

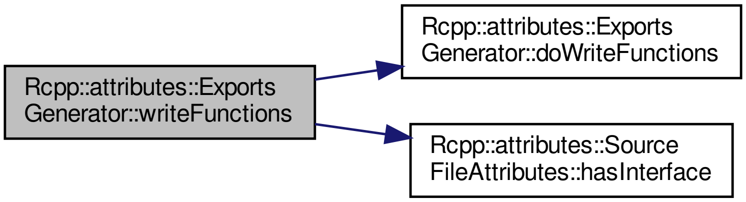
6.212.3.21 writeFunctions()

```
void Rcpp::attributes::ExportsGenerator::writeFunctions (
             const SourceFileAttributes & attributes,
             bool verbose )
```

Definition at line 1892 of file attributes.cpp.

References doWriteFunctions(), hasCppInterface_, Rcpp::attributes::SourceFileAttributes::hasInterface(), and Rcpp↩ ::attributes::kInterfaceCpp.

Here is the call graph for this function:


Rcpp::attributes::Exports
Generator::writeFunctions
Rcpp::attributes::Exports
Generator::doWriteFunctions
Rcpp::attributes::Source
FileAttributes::hasInterface

6.212.4 Member Data Documentation

6.212.4.1 codeStream_

```
std::ostringstream Rcpp::attributes::ExportsGenerator::codeStream_  [private]
```

Definition at line 691 of file attributes.cpp.

Referenced by commit(), operator std::ostream &(), and ostr().

6.212.4.2 commentPrefix_

```
std::string Rcpp::attributes::ExportsGenerator::commentPrefix_  [private]
```

Definition at line 689 of file attributes.cpp.

Referenced by commit().

6.212.4.3 existingCode_

```
std::string Rcpp::attributes::ExportsGenerator::existingCode_  [private]
```

Definition at line 690 of file attributes.cpp.

Referenced by commit(), ExportsGenerator(), and isSafeToOverwrite().

6.212.4.4 hasCppInterface_

```
bool Rcpp::attributes::ExportsGenerator::hasCppInterface_  [private]
```

Definition at line 692 of file attributes.cpp.

Referenced by hasCppInterface(), and writeFunctions().

6.212.4.5 package_

```
std::string Rcpp::attributes::ExportsGenerator::package_  [private]
```

Definition at line 687 of file attributes.cpp.

Referenced by package().

6.212.4.6 packageCpp_

```
std::string Rcpp::attributes::ExportsGenerator::packageCpp_  [private]
```

Definition at line 688 of file attributes.cpp.

Referenced by ExportsGenerator(), and packageCpp().

6.212.4.7 targetFile_

```
std::string Rcpp::attributes::ExportsGenerator::targetFile_  [private]
```

Definition at line 686 of file attributes.cpp.

Referenced by commit(), ExportsGenerator(), remove(), and targetFile().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.213 Rcpp::attributes::ExportsGenerators Class Reference

Public Types

- typedef std::vector< ExportsGenerator ∗ >::iterator Itr

Public Member Functions

- ExportsGenerators ()
- virtual ∼ExportsGenerators ()
- void add (ExportsGenerator ∗pGenerator)
- void writeBegin ()
- void writeFunctions (const SourceFileAttributes &attributes, bool verbose)
- void writeEnd (bool hasPackageInit)
- std::vector< std::string > commit (const std::vector< std::string > &includes)
- std::vector< std::string > remove ()

Private Member Functions

- ExportsGenerators (const ExportsGenerators &)
- ExportsGenerators & operator= (const ExportsGenerators &)

Private Attributes

- std::vector< ExportsGenerator ∗ > generators_

6.213.1 Detailed Description

Definition at line 790 of file attributes.cpp.

6.213.2 Member Typedef Documentation

6.213.2.1 Itr

```
typedef std::vector<ExportsGenerator*>::iterator Rcpp::attributes::ExportsGenerators::Itr
```

Definition at line 792 of file attributes.cpp.

6.213.3 Constructor & Destructor Documentation

6.213.3.1 ExportsGenerators() [1/2]

```
Rcpp::attributes::ExportsGenerators::ExportsGenerators ( )  [inline]
```

Definition at line 794 of file attributes.cpp.

6.213.3.2 ∼ExportsGenerators()

```
Rcpp::attributes::ExportsGenerators::∼ExportsGenerators ( )  [virtual]
```

Definition at line 2553 of file attributes.cpp.

References generators_.

6.213.3.3 ExportsGenerators() [2/2]

```
Rcpp::attributes::ExportsGenerators::ExportsGenerators (
            const ExportsGenerators &  )  [private]
```

6.213.4 Member Function Documentation

6.213.4.1 add()

```
void Rcpp::attributes::ExportsGenerators::add (
            ExportsGenerator * pGenerator )
```

Definition at line 2562 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.4.2 commit()

```
std::vector< std::string > Rcpp::attributes::ExportsGenerators::commit (
            const std::vector< std::string > & includes )
```

Definition at line 2584 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.4.3 operator=()

```
ExportsGenerators& Rcpp::attributes::ExportsGenerators::operator= (
            const ExportsGenerators & ) [private]
```

6.213.4.4 remove()

```
std::vector< std::string > Rcpp::attributes::ExportsGenerators::remove ( )
```

Definition at line 2598 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.4.5 writeBegin()

```
void Rcpp::attributes::ExportsGenerators::writeBegin ( )
```

Definition at line 2566 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.4.6 writeEnd()

```
void Rcpp::attributes::ExportsGenerators::writeEnd (
             bool hasPackageInit )
```

Definition at line 2578 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.4.7 writeFunctions()

```
void Rcpp::attributes::ExportsGenerators::writeFunctions (
             const SourceFileAttributes & attributes,
             bool verbose )
```

Definition at line 2571 of file attributes.cpp.

References generators_.

Referenced by compileAttributes().

6.213.5 Member Data Documentation

6.213.5.1 generators_

```
std::vector<ExportsGenerator*> Rcpp::attributes::ExportsGenerators::generators_  [private]
```

Definition at line 817 of file attributes.cpp.

Referenced by add(), commit(), remove(), writeBegin(), writeEnd(), writeFunctions(), and ∼ExportsGenerators().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.214 Rcpp::traits::Extractor< RTYPE, NA, VECTOR > Struct Template Reference

```
#include <Extractor.h>
```

Public Types

- typedef VECTOR type

6.214.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
struct Rcpp::traits::Extractor< RTYPE, NA, VECTOR >

Definition at line 29 of file Extractor.h.

6.214.2 Member Typedef Documentation

6.214.2.1 type

```
template<int RTYPE, bool NA, typename VECTOR >
typedef VECTOR Rcpp::traits::Extractor< RTYPE, NA, VECTOR >::type
```

Definition at line 30 of file Extractor.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/Extractor.h

6.215 tinyformat::detail::is_convertible< T1, T2 >::fail Struct Reference

Public Attributes

- char dummy [2]

6.215.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::fail

Definition at line 180 of file tinyformat.h.

6.215.2 Member Data Documentation

6.215.2.1 dummy

```
template<typename T1 , typename T2 >
char tinyformat::detail::is_convertible< T1, T2 >::fail::dummy[2]
```

Definition at line 180 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.216 Rcpp::Fast< VECTOR > Class Template Reference

```
#include <Fast.h>
```

Public Types

- typedef VECTOR::stored_type value_type

Public Member Functions

- Fast (VECTOR &v_)
- value_type & operator[ ] (R_xlen_t i)
- const value_type & operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- VECTOR & v
- value_type ∗ data

6.216.1 Detailed Description

template<typename VECTOR>
class Rcpp::Fast< VECTOR >

Definition at line 27 of file Fast.h.

6.216.2 Member Typedef Documentation

6.216.2.1 value_type

```
template<typename VECTOR >
typedef VECTOR::stored_type Rcpp::Fast< VECTOR >::value_type
```

Definition at line 29 of file Fast.h.

6.216.3 Constructor & Destructor Documentation

6.216.3.1 Fast()

```
template<typename VECTOR >
Rcpp::Fast< VECTOR >::Fast (
            VECTOR & v_ )  [inline]
```

Definition at line 31 of file Fast.h.

6.216.4 Member Function Documentation

6.216.4.1 operator[]() [1/2]

```
template<typename VECTOR >
value_type& Rcpp::Fast< VECTOR >::operator[] (
            R_xlen_t i )  [inline]
```

Definition at line 33 of file Fast.h.

References Rcpp::Fast< VECTOR >::data.

6.216.4.2 operator[]() [2/2]

```
template<typename VECTOR >
const value_type& Rcpp::Fast< VECTOR >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 34 of file Fast.h.

References Rcpp::Fast< VECTOR >::data.

6.216.4.3 size()

```
template<typename VECTOR >
R_xlen_t Rcpp::Fast< VECTOR >::size ( ) const  [inline]
```

Definition at line 35 of file Fast.h.

References Rcpp::Fast< VECTOR >::v.

6.216.5 Member Data Documentation

6.216.5.1 data

```
template<typename VECTOR >
value_type* Rcpp::Fast< VECTOR >::data  [private]
```

Definition at line 39 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::operator[ ]().

6.216.5.2 v

```
template<typename VECTOR >
VECTOR& Rcpp::Fast< VECTOR >::v  [private]
```

Definition at line 38 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::size().

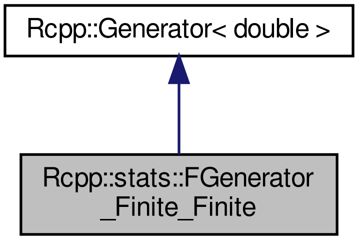
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Fast.h

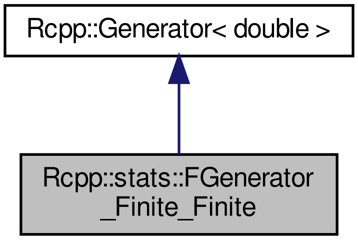
6.217 Rcpp::stats::FGenerator_Finite_Finite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator_Finite_Finite:


Rcpp::Generator< double >
Rcpp::stats::FGenerator_Finite_Finite

Collaboration diagram for Rcpp::stats::FGenerator_Finite_Finite:


Rcpp::Generator< double >
Rcpp::stats::FGenerator_Finite_Finite

Public Member Functions

- FGenerator_Finite_Finite (double n1_, double n2_)
- double operator() () const

Private Attributes

- double n1__2
- double n2__2
- double ratio

Additional Inherited Members

6.217.1 Detailed Description

Definition at line 28 of file rf.h.

6.217.2 Constructor & Destructor Documentation

6.217.2.1 FGenerator_Finite_Finite()

```
Rcpp::stats::FGenerator_Finite_Finite::FGenerator_Finite_Finite (
            double n1_,
            double n2_ )  [inline]
```

Definition at line 31 of file rf.h.

6.217.3 Member Function Documentation

6.217.3.1 operator()()

```
double Rcpp::stats::FGenerator_Finite_Finite::operator() ( ) const  [inline]
```

Definition at line 34 of file rf.h.

References n1__2, n2__2, and ratio.

6.217.4 Member Data Documentation

6.217.4.1 n1__2

```
double Rcpp::stats::FGenerator_Finite_Finite::n1__2  [private]
```

Definition at line 41 of file rf.h.

Referenced by operator()().

6.217.4.2 n2__2

```
double Rcpp::stats::FGenerator_Finite_Finite::n2__2  [private]
```

Definition at line 41 of file rf.h.

Referenced by operator()().

6.217.4.3 ratio

```
double Rcpp::stats::FGenerator_Finite_Finite::ratio  [private]
```

Definition at line 41 of file rf.h.

Referenced by operator()().

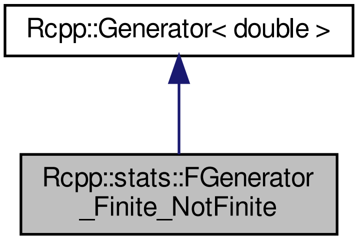
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

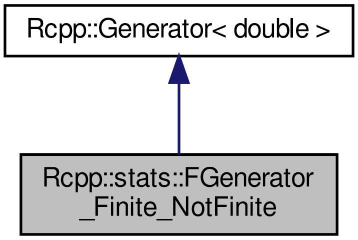
6.218 Rcpp::stats::FGenerator_Finite_NotFinite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator_Finite_NotFinite:



Collaboration diagram for Rcpp::stats::FGenerator_Finite_NotFinite:



Public Member Functions

- FGenerator_Finite_NotFinite (double n1_)
- double operator() () const

Private Attributes

- double n1
- double n1__2

Additional Inherited Members

6.218.1 Detailed Description

Definition at line 58 of file rf.h.

6.218.2 Constructor & Destructor Documentation

6.218.2.1 FGenerator_Finite_NotFinite()

```
Rcpp::stats::FGenerator_Finite_NotFinite::FGenerator_Finite_NotFinite (
             double n1_ )  [inline]
```

Definition at line 61 of file rf.h.

6.218.3 Member Function Documentation

6.218.3.1 operator()()

```
double Rcpp::stats::FGenerator_Finite_NotFinite::operator() ( ) const  [inline]
```

Definition at line 63 of file rf.h.

References n1, and n1__2.

6.218.4 Member Data Documentation

6.218.4.1 n1

```
double Rcpp::stats::FGenerator_Finite_NotFinite::n1  [private]
```

Definition at line 69 of file rf.h.

Referenced by operator()().

6.218.4.2 n1__2

```
double Rcpp::stats::FGenerator_Finite_NotFinite::n1__2  [private]
```

Definition at line 69 of file rf.h.

Referenced by operator()().

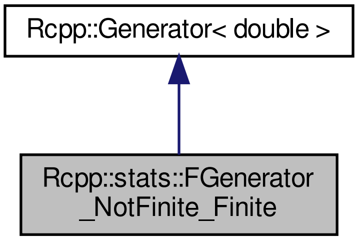
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

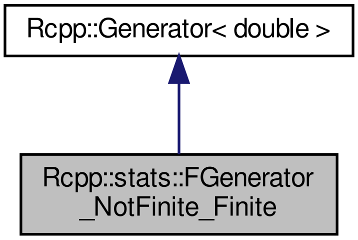
6.219 Rcpp::stats::FGenerator_NotFinite_Finite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator_NotFinite_Finite:



Collaboration diagram for Rcpp::stats::FGenerator_NotFinite_Finite:



Public Member Functions

- FGenerator_NotFinite_Finite (double n2_)
- double operator() () const

Private Attributes

- double n2
- double n2__2

Additional Inherited Members

6.219.1 Detailed Description

Definition at line 44 of file rf.h.

6.219.2 Constructor & Destructor Documentation

6.219.2.1 FGenerator_NotFinite_Finite()

```
Rcpp::stats::FGenerator_NotFinite_Finite::FGenerator_NotFinite_Finite (
             double n2_ )  [inline]
```

Definition at line 47 of file rf.h.

6.219.3 Member Function Documentation

6.219.3.1 operator()()

```
double Rcpp::stats::FGenerator_NotFinite_Finite::operator() ( ) const  [inline]
```

Definition at line 49 of file rf.h.

References n2, and n2__2.

6.219.4 Member Data Documentation

6.219.4.1 n2

```
double Rcpp::stats::FGenerator_NotFinite_Finite::n2  [private]
```

Definition at line 55 of file rf.h.

Referenced by operator()().

6.219.4.2 n2__2

```
double Rcpp::stats::FGenerator_NotFinite_Finite::n2__2  [private]
```

Definition at line 55 of file rf.h.

Referenced by operator()().

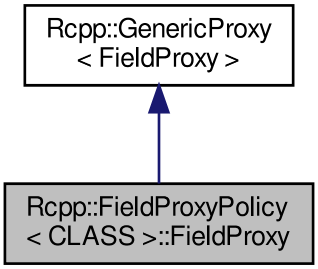
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

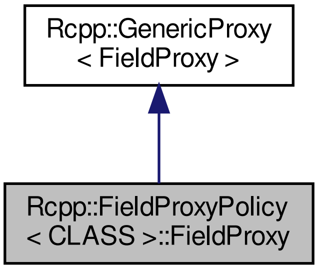
6.220 Rcpp::FieldProxyPolicy< CLASS >::FieldProxy Class Reference

```
#include <FieldProxy.h>
```

Inheritance diagram for Rcpp::FieldProxyPolicy< CLASS >::FieldProxy:



Collaboration diagram for Rcpp::FieldProxyPolicy< CLASS >::FieldProxy:



Public Member Functions

- FieldProxy (CLASS &v, const std::string &name)
- FieldProxy & operator= (const FieldProxy &rhs)
- template<typename T >
  FieldProxy & operator= (const T &rhs)
- template<typename T >
  operator T () const
- operator SEXP () const
- template<typename CLASS >
  FieldProxyPolicy< CLASS >::FieldProxy & operator= (const FieldProxyPolicy< CLASS >::FieldProxy &rhs)
- template<typename T >
  FieldProxyPolicy< CLASS >::FieldProxy & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- CLASS & parent
- const std::string & field_name

6.220.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >::FieldProxy

Definition at line 27 of file FieldProxy.h.

6.220.2 Constructor & Destructor Documentation

6.220.2.1 FieldProxy()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::FieldProxy (
             CLASS & v,
             const std::string & name )  [inline]
```

Definition at line 29 of file FieldProxy.h.

6.220.3 Member Function Documentation

6.220.3.1 get()

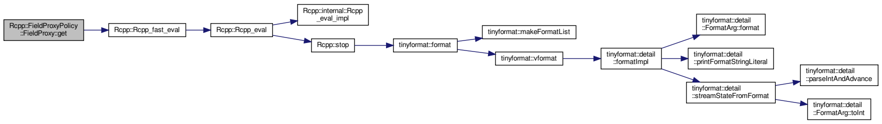
```
template<typename CLASS >
SEXP Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get ( ) const  [inline], [private]
```

Definition at line 43 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field_name, Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent, and Rcpp::Rcpp_fast_eval().

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator SEXP(), and Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=().

Here is the call graph for this function:



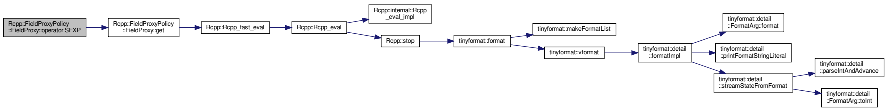
6.220.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator SEXP ( ) const  [inline]
```

Definition at line 37 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get().

Here is the call graph for this function:



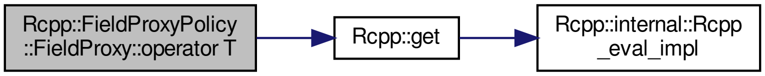
6.220.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator T
```

Definition at line 193 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



6.220.3.4 operator=() [1/4]

```
template<typename CLASS >
FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
             const FieldProxy & rhs )
```

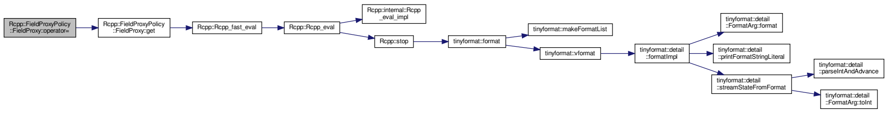
6.220.3.5 operator=() [2/4]

```
template<typename CLASS >
template<typename CLASS >
FieldProxyPolicy<CLASS>::FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
             const FieldProxyPolicy< CLASS >::FieldProxy & rhs )
```

Definition at line 178 of file proxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get().

Here is the call graph for this function:



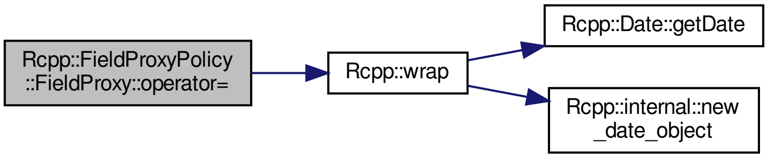
6.220.3.6 operator=() [3/4]

```
template<typename CLASS >
template<typename T >
FieldProxyPolicy<CLASS>::FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
            const T & rhs )
```

Definition at line 186 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::FieldProxyPolicy ::FieldProxy::operator=
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.220.3.7 operator=() [4/4]

```
template<typename CLASS >
template<typename T >
FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
            const T & rhs )
```

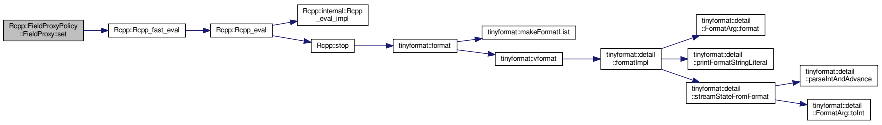
6.220.3.8 set()

```
template<typename CLASS >
void Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set (
            SEXP x ) [inline], [private]
```

Definition at line 48 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field_name, Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent, and Rcpp::Rcpp_fast_eval().

Here is the call graph for this function:


Rcpp::FieldProxyPolicy ::FieldProxy::set
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.220.4 Member Data Documentation

6.220.4.1 field_name

```
template<typename CLASS >
const std::string& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field_name  [private]
```

Definition at line 41 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get(), and Rcpp::FieldProxyPolicy< CLASS >::Field↩ Proxy::set().

6.220.4.2 parent

```
template<typename CLASS >
CLASS& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent  [private]
```

Definition at line 40 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get(), and Rcpp::FieldProxyPolicy< CLASS >::Field↩ Proxy::set().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/FieldProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.221 Rcpp::FieldProxyPolicy< CLASS > Class Template Reference

```
#include <FieldProxy.h>
```

Classes

- class const_FieldProxy
- class FieldProxy

Public Member Functions

- FieldProxy field (const std::string &name)
- const_FieldProxy field (const std::string &name) const

6.221.1 Detailed Description

template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >

Definition at line 24 of file FieldProxy.h.

6.221.2 Member Function Documentation

6.221.2.1 field() [1/2]

```
template<typename CLASS >
FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (
             const std::string & name )  [inline]
```

Definition at line 77 of file FieldProxy.h.

6.221.2.2 field() [2/2]

```
template<typename CLASS >
const_FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (
             const std::string & name ) const  [inline]
```

Definition at line 80 of file FieldProxy.h.

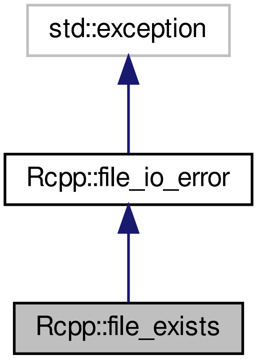
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/FieldProxy.h

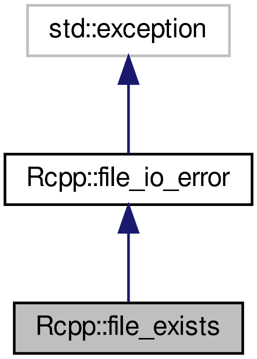
6.222 Rcpp::file_exists Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file_exists:



Collaboration diagram for Rcpp::file_exists:



Public Member Functions

- file_exists (const std::string &file) throw ()

6.222.1 Detailed Description

Definition at line 104 of file exceptions.h.

6.222.2 Constructor & Destructor Documentation

6.222.2.1 file_exists()

```
Rcpp::file_exists::file_exists (
            const std::string & file ) throw ( )   [inline]
```

Definition at line 106 of file exceptions.h.

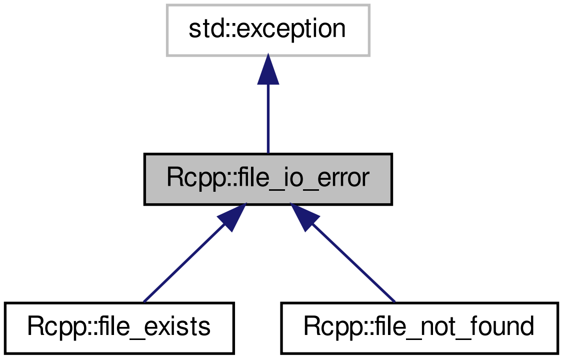
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

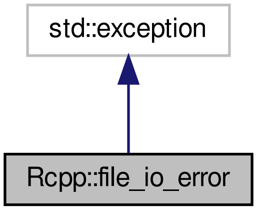
6.223 Rcpp::file_io_error Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file_io_error:



Collaboration diagram for Rcpp::file_io_error:



Public Member Functions

- file_io_error (const std::string &file) throw ()
- file_io_error (int code, const std::string &file) throw ()
- file_io_error (const std::string &msg, const std::string &file) throw ()
- virtual ∼file_io_error () throw ()
- virtual const char ∗ what () const throw ()
- std::string filePath () const throw ()

Private Attributes

- std::string message
- std::string file

6.223.1 Detailed Description

Definition at line 82 of file exceptions.h.

6.223.2 Constructor & Destructor Documentation

6.223.2.1 file_io_error() [1/3]

```
Rcpp::file_io_error::file_io_error (
            const std::string & file ) throw ( )   [inline]
```

Definition at line 84 of file exceptions.h.

6.223.2.2 file_io_error() [2/3]

```
Rcpp::file_io_error::file_io_error (
            int code,
            const std::string & file ) throw ( )   [inline]
```

Definition at line 86 of file exceptions.h.

6.223.2.3 file_io_error() [3/3]

```
Rcpp::file_io_error::file_io_error (
            const std::string & msg,
            const std::string & file ) throw ( )   [inline]
```

Definition at line 88 of file exceptions.h.

6.223.2.4 ∼file_io_error()

```
virtual Rcpp::file_io_error::∼file_io_error ( ) throw ( )   [inline], [virtual]
```

Definition at line 90 of file exceptions.h.

6.223.3 Member Function Documentation

6.223.3.1 filePath()

```
std::string Rcpp::file_io_error::filePath ( ) const throw ( )   [inline]
```

Definition at line 92 of file exceptions.h.

References file.

Referenced by compileAttributes().

6.223.3.2 what()

```
virtual const char* Rcpp::file_io_error::what ( ) const throw ( )   [inline], [virtual]
```

Definition at line 91 of file exceptions.h.

References message.

6.223.4 Member Data Documentation

6.223.4.1 file

```
std::string Rcpp::file_io_error::file  [private]
```

Definition at line 95 of file exceptions.h.

Referenced by filePath().

6.223.4.2 message

```
std::string Rcpp::file_io_error::message  [private]
```

Definition at line 94 of file exceptions.h.

Referenced by what().

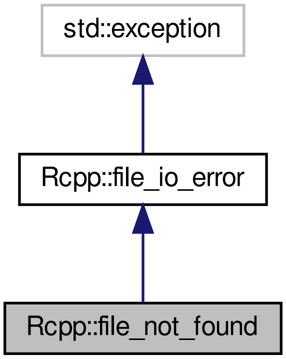
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

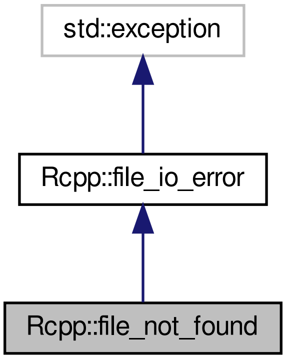
6.224 Rcpp::file_not_found Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file_not_found:



Collaboration diagram for Rcpp::file_not_found:



Public Member Functions

- file_not_found (const std::string &file) throw ()

6.224.1 Detailed Description

Definition at line 98 of file exceptions.h.

6.224.2 Constructor & Destructor Documentation

6.224.2.1 file_not_found()

```
Rcpp::file_not_found::file_not_found (
             const std::string & file ) throw ( )   [inline]
```

Definition at line 100 of file exceptions.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

6.225 Rcpp::attributes::FileInfo Class Reference

Public Member Functions

- FileInfo (const std::string &path)
- FileInfo (const List &fileInfo)
- List toList () const
- std::string path () const
- bool exists () const
- double lastModified () const
- std::string extension () const
- bool operator< (const FileInfo &other) const
- bool operator== (const FileInfo &other) const
- bool operator!= (const FileInfo &other) const
- std::ostream & operator<< (std::ostream &os) const

Private Attributes

- std::string path_
- bool exists_
- double lastModified_

6.225.1 Detailed Description

Definition at line 49 of file attributes.cpp.

6.225.2 Constructor & Destructor Documentation

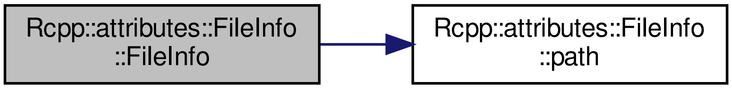
6.225.2.1 FileInfo() [1/2]

```
Rcpp::attributes::FileInfo::FileInfo (
             const std::string & path )  [explicit]
```

Definition at line 2980 of file attributes.cpp.

References exists_, lastModified_, and path().

Here is the call graph for this function:



6.225.2.2 FileInfo() [2/2]

```
Rcpp::attributes::FileInfo::FileInfo (
             const List & fileInfo )  [inline], [explicit]
```

Definition at line 56 of file attributes.cpp.

References exists_, lastModified_, and path_.

6.225.3 Member Function Documentation

6.225.3.1 exists()

```
bool Rcpp::attributes::FileInfo::exists ( ) const  [inline]
```

Definition at line 72 of file attributes.cpp.

References exists_.

6.225.3.2 extension()

```
std::string Rcpp::attributes::FileInfo::extension ( ) const  [inline]
```

Definition at line 75 of file attributes.cpp.

References path_.

6.225.3.3 lastModified()

```
double Rcpp::attributes::FileInfo::lastModified ( ) const  [inline]
```

Definition at line 73 of file attributes.cpp.

References lastModified_.

6.225.3.4 operator"!=()

```
bool Rcpp::attributes::FileInfo::operator!= (
             const FileInfo & other ) const  [inline]
```

Definition at line 93 of file attributes.cpp.

6.225.3.5 operator<()

```
bool Rcpp::attributes::FileInfo::operator< (
             const FileInfo & other ) const  [inline]
```

Definition at line 83 of file attributes.cpp.

References path_.

6.225.3.6 operator<<()

```
std::ostream& Rcpp::attributes::FileInfo::operator<< (
             std::ostream & os ) const  [inline]
```

Definition at line 97 of file attributes.cpp.

References path_.

6.225.3.7 operator==()

```
bool Rcpp::attributes::FileInfo::operator== (
             const FileInfo & other ) const  [inline]
```

Definition at line 87 of file attributes.cpp.

References exists_, lastModified_, and path_.

6.225.3.8 path()

```
std::string Rcpp::attributes::FileInfo::path ( ) const  [inline]
```

Definition at line 71 of file attributes.cpp.

References path_.

Referenced by FileInfo().

6.225.3.9 toList()

```
List Rcpp::attributes::FileInfo::toList ( ) const  [inline]
```

Definition at line 63 of file attributes.cpp.

References exists_, lastModified_, and path_.

6.225.4 Member Data Documentation

6.225.4.1 exists_

```
bool Rcpp::attributes::FileInfo::exists_  [private]
```

Definition at line 104 of file attributes.cpp.

Referenced by exists(), FileInfo(), operator==(), and toList().

6.225.4.2 lastModified_

```
double Rcpp::attributes::FileInfo::lastModified_  [private]
```

Definition at line 105 of file attributes.cpp.

Referenced by FileInfo(), lastModified(), operator==(), and toList().

6.225.4.3 path_

```
std::string Rcpp::attributes::FileInfo::path_  [private]
```

Definition at line 103 of file attributes.cpp.

Referenced by extension(), FileInfo(), operator<(), operator<<(), operator==(), path(), and toList().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.226 Rcpp::fixed_call< RESULT_TYPE > Class Template Reference

```
#include <Language.h>
```

Public Types

- typedef RESULT_TYPE result_type

Public Member Functions

- fixed_call (Language call_)
- fixed_call (Function fun)
- RESULT_TYPE operator() ()

Private Attributes

- Language call

6.226.1 Detailed Description

template<typename RESULT_TYPE = SEXP>
class Rcpp::fixed_call< RESULT_TYPE >

Definition at line 163 of file Language.h.

6.226.2 Member Typedef Documentation

6.226.2.1 result_type

```
template<typename RESULT_TYPE = SEXP>
typedef RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::result_type
```

Definition at line 165 of file Language.h.

6.226.3 Constructor & Destructor Documentation

6.226.3.1 fixed_call() [1/2]

```
template<typename RESULT_TYPE = SEXP>
Rcpp::fixed_call< RESULT_TYPE >::fixed_call (
             Language call_ )  [inline]
```

Definition at line 167 of file Language.h.

6.226.3.2 fixed_call() [2/2]

```
template<typename RESULT_TYPE = SEXP>
Rcpp::fixed_call< RESULT_TYPE >::fixed_call (
             Function fun )  [inline]
```

Definition at line 168 of file Language.h.

6.226.4 Member Function Documentation

6.226.4.1 operator()()

```
template<typename RESULT_TYPE = SEXP>
RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::operator() ( )  [inline]
```

Definition at line 170 of file Language.h.

References Rcpp::fixed_call< RESULT_TYPE >::call.

6.226.5 Member Data Documentation

6.226.5.1 call

```
template<typename RESULT_TYPE = SEXP>
Language Rcpp::fixed_call< RESULT_TYPE >::call  [private]
```

Definition at line 175 of file Language.h.

Referenced by Rcpp::fixed_call< RESULT_TYPE >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.227 Rcpp::sugar::forbidden_conversion< bool > Class Template Reference

6.227.1 Detailed Description

template< bool >
class Rcpp::sugar::forbidden_conversion< bool >

Definition at line 29 of file SingleLogicalResult.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

6.228 Rcpp::sugar::forbidden_conversion< true > Class Reference

```
#include <SingleLogicalResult.h>
```

6.228.1 Detailed Description

Definition at line 32 of file SingleLogicalResult.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

6.229 tinyformat::detail::FormatArg Class Reference

```
#include <tinyformat.h>
```

Public Member Functions

- FormatArg ()
- 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)

6.229.1 Detailed Description

Definition at line 492 of file tinyformat.h.

6.229.2 Constructor & Destructor Documentation

6.229.2.1 FormatArg() [1/2]

```
tinyformat::detail::FormatArg::FormatArg ( )  [inline]
```

Definition at line 495 of file tinyformat.h.

6.229.2.2 FormatArg() [2/2]

```
template<typename T >
tinyformat::detail::FormatArg::FormatArg (
             const T & value )  [inline]
```

Definition at line 502 of file tinyformat.h.

6.229.3 Member Function Documentation

6.229.3.1 format()

```
void tinyformat::detail::FormatArg::format (
             std::ostream & out,
             const char * fmtBegin,
             const char * fmtEnd,
             int ntrunc ) const  [inline]
```

Definition at line 508 of file tinyformat.h.

References m_formatImpl, m_value, and TINYFORMAT_ASSERT.

Referenced by tinyformat::detail::formatImpl().

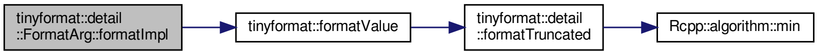
6.229.3.2 formatImpl()

```
template<typename T >
static TINYFORMAT_HIDDEN void tinyformat::detail::FormatArg::formatImpl (
             std::ostream & out,
             const char * fmtBegin,
             const char * fmtEnd,
             int ntrunc,
             const void * value )  [inline], [static], [private]
```

Definition at line 525 of file tinyformat.h.

References tinyformat::formatValue().

Here is the call graph for this function:


tinyformat::detail ::FormatArg::formatImpl
tinyformat::formatValue
tinyformat::detail ::formatTruncated
Rcpp::algorithm::min

6.229.3.3 toInt()

```
int tinyformat::detail::FormatArg::toInt ( ) const  [inline]
```

Definition at line 516 of file tinyformat.h.

References m_toIntImpl, m_value, and TINYFORMAT_ASSERT.

Referenced by tinyformat::detail::streamStateFromFormat().

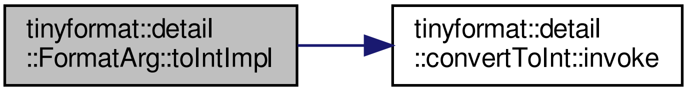
6.229.3.4 toIntImpl()

```
template<typename T >
static TINYFORMAT_HIDDEN int tinyformat::detail::FormatArg::toIntImpl (
             const void * value )  [inline], [static], [private]
```

Definition at line 532 of file tinyformat.h.

References tinyformat::detail::convertToInt< T, convertible >::invoke().

Here is the call graph for this function:


tinyformat::detail ::FormatArg::toIntImpl
tinyformat::detail ::convertToInt::invoke

6.229.4 Member Data Documentation

6.229.4.1 m_formatImpl

```
void(* tinyformat::detail::FormatArg::m_formatImpl) (std::ostream &out, const char *fmtBegin,
const char *fmtEnd, int ntrunc, const void *value)  [private]
```

Definition at line 538 of file tinyformat.h.

Referenced by format().

6.229.4.2 m_toIntImpl

```
int(* tinyformat::detail::FormatArg::m_toIntImpl) (const void *value)  [private]
```

Definition at line 540 of file tinyformat.h.

Referenced by toInt().

6.229.4.3 m_value

```
const void* tinyformat::detail::FormatArg::m_value  [private]
```

Definition at line 537 of file tinyformat.h.

Referenced by format(), and toInt().

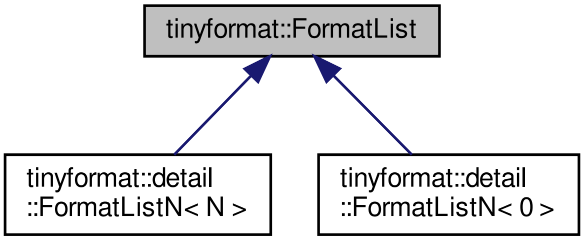
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

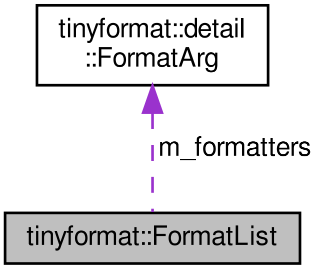
6.230 tinyformat::FormatList Class Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::FormatList:


tinyformat::FormatList
tinyformat::detail ::FormatListN< N >
tinyformat::detail ::FormatListN< 0 >

Collaboration diagram for tinyformat::FormatList:



Public Member Functions

- FormatList (detail::FormatArg ∗formatters, int N)

Private Attributes

- const detail::FormatArg ∗ m_formatters
- int m_N

Friends

- void vformat (std::ostream &out, const char ∗fmt, const FormatList &list)

6.230.1 Detailed Description

List of template arguments format(), held in a type-opaque way.

A const reference to FormatList (typedef'd as FormatListRef) may be conveniently used to pass arguments to non-template functions: All type information has been stripped from the arguments, leaving just enough of a common interface to perform formatting as required.

Definition at line 852 of file tinyformat.h.

6.230.2 Constructor & Destructor Documentation

6.230.2.1 FormatList()

```
tinyformat::FormatList::FormatList (
            detail::FormatArg ∗ formatters,
            int N )  [inline]
```

Definition at line 855 of file tinyformat.h.

6.230.3 Friends And Related Function Documentation

6.230.3.1 vformat

```
void vformat (
            std::ostream & out,
            const char ∗ fmt,
            const FormatList & list )  [friend]
```

6.230.4 Member Data Documentation

6.230.4.1 m_formatters

```
const detail::FormatArg* tinyformat::FormatList::m_formatters  [private]
```

Definition at line 862 of file tinyformat.h.

Referenced by tinyformat::vformat().

6.230.4.2 m_N

```
int tinyformat::FormatList::m_N  [private]
```

Definition at line 863 of file tinyformat.h.

Referenced by tinyformat::vformat().

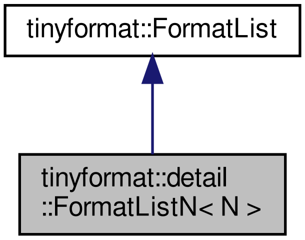
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

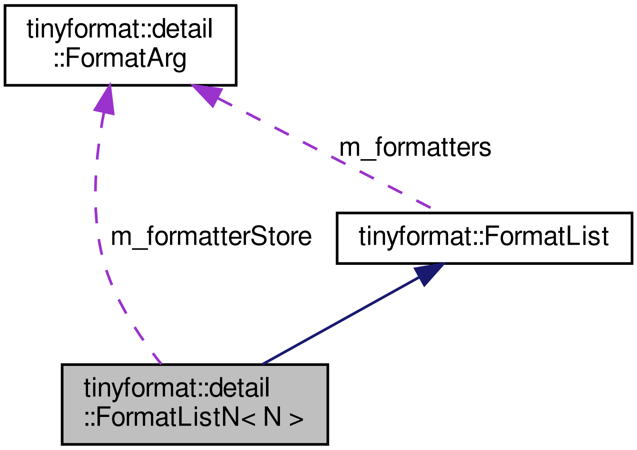
6.231 tinyformat::detail::FormatListN< N > Class Template Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::detail::FormatListN< N >:



Collaboration diagram for tinyformat::detail::FormatListN< N >:



Public Member Functions

- void init (int)

Private Attributes

- FormatArg m_formatterStore [N]

6.231.1 Detailed Description

template<int N>
class tinyformat::detail::FormatListN< N >

Definition at line 874 of file tinyformat.h.

6.231.2 Member Function Documentation

6.231.2.1 init()

```
template<int N>
void tinyformat::detail::FormatListN< N >::init (
             int  )  [inline]
```

Definition at line 884 of file tinyformat.h.

6.231.3 Member Data Documentation

6.231.3.1 m_formatterStore

```
template<int N>
FormatArg tinyformat::detail::FormatListN< N >::m_formatterStore[N]  [private]
```

Definition at line 904 of file tinyformat.h.

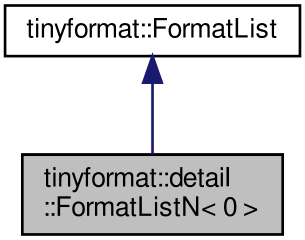
The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

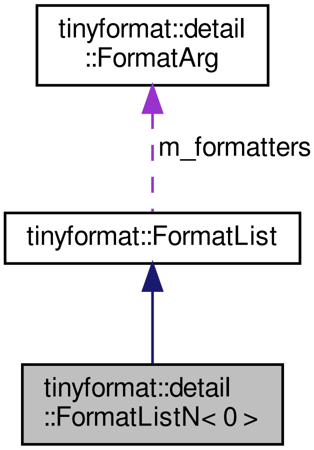
6.232 tinyformat::detail::FormatListN< 0 > Class Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::detail::FormatListN< 0 >:



Collaboration diagram for tinyformat::detail::FormatListN< 0 >:



Public Member Functions

- FormatListN ()

6.232.1 Detailed Description

Definition at line 908 of file tinyformat.h.

6.232.2 Constructor & Destructor Documentation

6.232.2.1 FormatListN()

```
tinyformat::detail::FormatListN< 0 >::FormatListN ( )  [inline]
```

Definition at line 910 of file tinyformat.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.233 tinyformat::detail::formatValueAsType< T, fmtT, convertible > Struct Template Reference

```
#include <tinyformat.h>
```

Static Public Member Functions

- static void invoke (std::ostream &, const T &)

6.233.1 Detailed Description

template<typename T, typename fmtT, bool convertible = is_convertible<T, fmtT>::value>
struct tinyformat::detail::formatValueAsType< T, fmtT, convertible >

Definition at line 216 of file tinyformat.h.

6.233.2 Member Function Documentation

6.233.2.1 invoke()

```
template<typename T , typename fmtT , bool convertible = is_convertible<T, fmtT>::value>
static void tinyformat::detail::formatValueAsType< T, fmtT, convertible >::invoke (
             std::ostream & ,
             const T &  )  [inline], [static]
```

Definition at line 218 of file tinyformat.h.

References TINYFORMAT_ASSERT.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.234 tinyformat::detail::formatValueAsType< T, fmtT, true > Struct Template Reference

```
#include <tinyformat.h>
```

Static Public Member Functions

- static void invoke (std::ostream &out, const T &value)

6.234.1 Detailed Description

template<typename T, typename fmtT>
struct tinyformat::detail::formatValueAsType< T, fmtT, true >

Definition at line 223 of file tinyformat.h.

6.234.2 Member Function Documentation

6.234.2.1 invoke()

```
template<typename T , typename fmtT >
static void tinyformat::detail::formatValueAsType< T, fmtT, true >::invoke (
             std::ostream & out,
             const T & value )  [inline], [static]
```

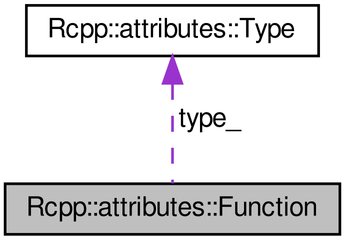
Definition at line 225 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.235 Rcpp::attributes::Function Class Reference

Collaboration diagram for Rcpp::attributes::Function:



Public Member Functions

- Function ()
- Function (const Type &type, const std::string &name, const std::vector< Argument > &arguments)
- Function renamedTo (const std::string &name) const
- std::string signature () const
- std::string signature (const std::string &name) const
- bool isHidden () const
- bool empty () const
- bool operator== (const Function &other) const
- bool operator!= (const Function &other) const
- const Type & type () const
- const std::string & name () const
- const std::vector< Argument > & arguments () const

Private Attributes

- Type type_
- std::string name_
- std::vector< Argument > arguments_

6.235.1 Detailed Description

Definition at line 245 of file attributes.cpp.

6.235.2 Constructor & Destructor Documentation

6.235.2.1 Function() [1/2]

```
Rcpp::attributes::Function::Function ( )  [inline]
```

Definition at line 247 of file attributes.cpp.

Referenced by renamedTo().

6.235.2.2 Function() [2/2]

```
Rcpp::attributes::Function::Function (
             const Type & type,
             const std::string & name,
             const std::vector< Argument > & arguments )  [inline]
```

Definition at line 248 of file attributes.cpp.

6.235.3 Member Function Documentation

6.235.3.1 arguments()

```
const std::vector<Argument>& Rcpp::attributes::Function::arguments ( ) const  [inline]
```

Definition at line 280 of file attributes.cpp.

References arguments_.

Referenced by renamedTo(), signature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

6.235.3.2 empty()

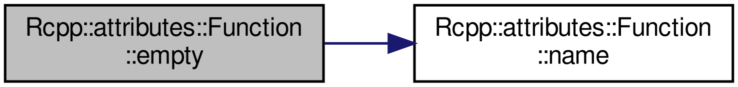
```
bool Rcpp::attributes::Function::empty ( ) const  [inline]
```

Definition at line 266 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::operator<<().

Here is the call graph for this function:


Rcpp::attributes::Function ::empty
Rcpp::attributes::Function ::name

6.235.3.3 isHidden()

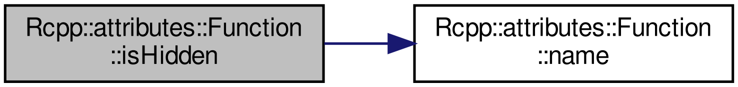
```
bool Rcpp::attributes::Function::isHidden ( ) const  [inline]
```

Definition at line 262 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

Here is the call graph for this function:



6.235.3.4 name()

```
const std::string& Rcpp::attributes::Function::name ( ) const  [inline]
```

Definition at line 279 of file attributes.cpp.

References name_.

Referenced by empty(), Rcpp::attributes::generateCpp(), isHidden(), renamedTo(), signature(), and Rcpp::attributes::↩
CppExportsGenerator::writeEnd().

6.235.3.5 operator"!=()

```
bool Rcpp::attributes::Function::operator!= (
             const Function & other ) const  [inline]
```

Definition at line 274 of file attributes.cpp.

6.235.3.6 operator==()

```
bool Rcpp::attributes::Function::operator== (
            const Function & other ) const  [inline]
```

Definition at line 268 of file attributes.cpp.

References arguments_, name_, and type_.

6.235.3.7 renamedTo()

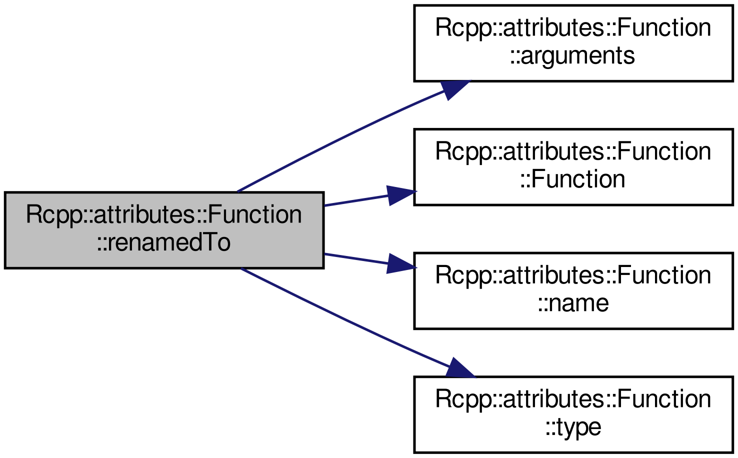
```
Function Rcpp::attributes::Function::renamedTo (
            const std::string & name ) const  [inline]
```

Definition at line 255 of file attributes.cpp.

References arguments(), Function(), name(), and type().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), and Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions().

Here is the call graph for this function:


Rcpp::attributes::Function ::renamedTo
Rcpp::attributes::Function ::arguments
Rcpp::attributes::Function ::Function
Rcpp::attributes::Function ::name
Rcpp::attributes::Function ::type

6.235.3.8 signature() `[1/2]`

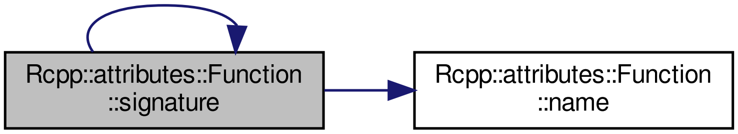
```
std::string Rcpp::attributes::Function::signature ( ) const  [inline]
```

Definition at line 259 of file attributes.cpp.

References name(), and signature().

Referenced by signature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



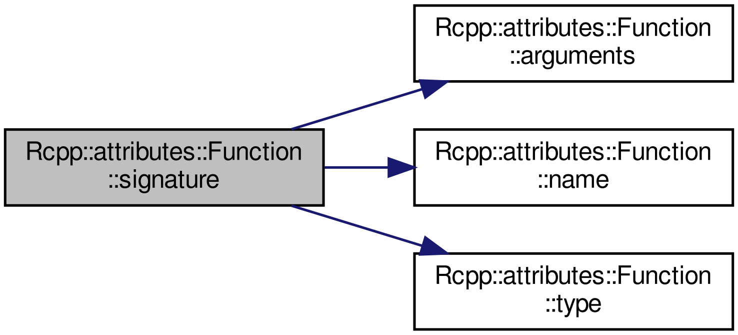
6.235.3.9 signature() `[2/2]`

```
std::string Rcpp::attributes::Function::signature (
             const std::string & name ) const
```

Definition at line 1042 of file attributes.cpp.

References arguments(), name(), and type().

Here is the call graph for this function:


Rcpp::attributes::Function
::arguments
Rcpp::attributes::Function
::signature
Rcpp::attributes::Function
::name
Rcpp::attributes::Function
::type

6.235.3.10 type()

```
const Type& Rcpp::attributes::Function::type ( ) const  [inline]
```

Definition at line 278 of file attributes.cpp.

References type_.

Referenced by renamedTo(), and signature().

6.235.4 Member Data Documentation

6.235.4.1 arguments_

```
std::vector<Argument> Rcpp::attributes::Function::arguments_  [private]
```

Definition at line 285 of file attributes.cpp.

Referenced by arguments(), and operator==().

6.235.4.2 name_

```
std::string Rcpp::attributes::Function::name_  [private]
```

Definition at line 284 of file attributes.cpp.

Referenced by name(), and operator==().

6.235.4.3 type_

```
Type Rcpp::attributes::Function::type_  [private]
```

Definition at line 283 of file attributes.cpp.

Referenced by operator==(), and type().

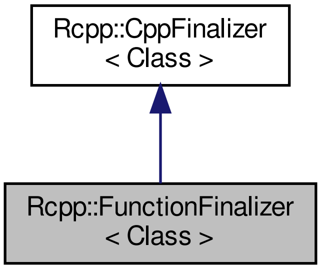
The documentation for this class was generated from the following file:

- src/attributes.cpp

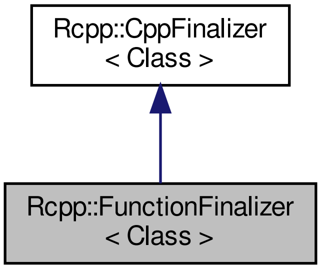
6.236 Rcpp::FunctionFinalizer< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::FunctionFinalizer< Class >:



Collaboration diagram for Rcpp::FunctionFinalizer< Class >:



Public Types

- typedef void(∗ Pointer) (Class ∗)

Public Member Functions

- FunctionFinalizer (Pointer p)
- virtual void run (Class ∗object)

Private Attributes

- Pointer finalizer

6.236.1 Detailed Description

template<typename Class>
class Rcpp::FunctionFinalizer< Class >

Definition at line 307 of file Module.h.

6.236.2 Member Typedef Documentation

6.236.2.1 Pointer

```
template<typename Class >
typedef void(* Rcpp::FunctionFinalizer< Class >::Pointer) (Class *)
```

Definition at line 309 of file Module.h.

6.236.3 Constructor & Destructor Documentation

6.236.3.1 FunctionFinalizer()

```
template<typename Class >
Rcpp::FunctionFinalizer< Class >::FunctionFinalizer (
             Pointer p )  [inline]
```

Definition at line 310 of file Module.h.

6.236.4 Member Function Documentation

6.236.4.1 run()

```
template<typename Class >
virtual void Rcpp::FunctionFinalizer< Class >::run (
             Class * object ) [inline], [virtual]
```

Reimplemented from Rcpp::CppFinalizer< Class >.

Definition at line 312 of file Module.h.

References Rcpp::FunctionFinalizer< Class >::finalizer.

6.236.5 Member Data Documentation

6.236.5.1 finalizer

```
template<typename Class >
Pointer Rcpp::FunctionFinalizer< Class >::finalizer [private]
```

Definition at line 317 of file Module.h.

Referenced by Rcpp::FunctionFinalizer< Class >::run().

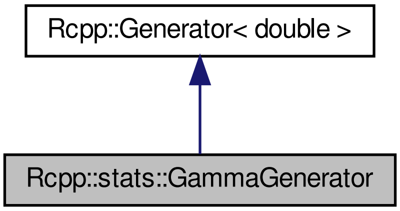
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

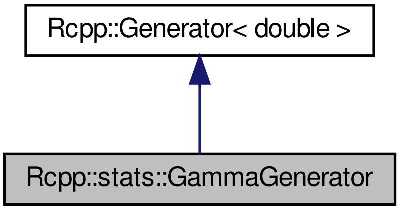
6.237 Rcpp::stats::GammaGenerator Class Reference

```
#include <rgamma.h>
```

Inheritance diagram for Rcpp::stats::GammaGenerator:



Collaboration diagram for Rcpp::stats::GammaGenerator:



Public Member Functions

- GammaGenerator (double a_, double scale_)
- double operator() () const

Private Attributes

- double a
- double scale

Additional Inherited Members

6.237.1 Detailed Description

Definition at line 28 of file rgamma.h.

6.237.2 Constructor & Destructor Documentation

6.237.2.1 GammaGenerator()

```
Rcpp::stats::GammaGenerator::GammaGenerator (
             double a_,
             double scale_ )  [inline]
```

Definition at line 30 of file rgamma.h.

6.237.3 Member Function Documentation

6.237.3.1 operator()()

```
double Rcpp::stats::GammaGenerator::operator() ( ) const  [inline]
```

Definition at line 32 of file rgamma.h.

References a, and scale.

6.237.4 Member Data Documentation

6.237.4.1 a

```
double Rcpp::stats::GammaGenerator::a  [private]
```

Definition at line 34 of file rgamma.h.

Referenced by operator()().

6.237.4.2 scale

```
double Rcpp::stats::GammaGenerator::scale  [private]
```

Definition at line 34 of file rgamma.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rgamma.h

6.238 Rcpp::Generator< T > Class Template Reference

```
#include <random.h>
```

Public Types

- typedef T r_generator

6.238.1 Detailed Description

template<typename T>
class Rcpp::Generator< T >

Definition at line 28 of file random.h.

6.238.2 Member Typedef Documentation

6.238.2.1 r_generator

```
template<typename T >
typedef T Rcpp::Generator< T >::r_generator
```

Definition at line 30 of file random.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/random.h

6.239 Rcpp::internal::generic_element_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

Public Types

- typedef SEXP target

Static Public Member Functions

- template<typename T >
  static SEXP get (const T &input)
- static SEXP get (const char ∗input)
- static SEXP get (SEXP input)

6.239.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::generic_element_converter< RTYPE >

Definition at line 74 of file converter.h.

6.239.2 Member Typedef Documentation

6.239.2.1 target

```
template<int RTYPE>
typedef SEXP Rcpp::internal::generic_element_converter< RTYPE >::target
```

Definition at line 76 of file converter.h.

6.239.3 Member Function Documentation

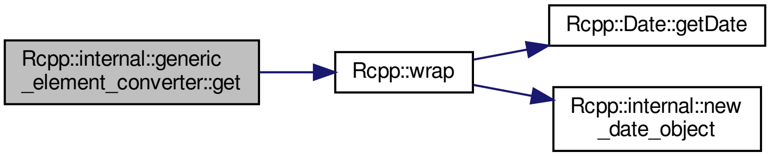
6.239.3.1 get() [1/3]

```
template<int RTYPE>
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
             const char * input )  [inline], [static]
```

Definition at line 83 of file converter.h.

References Rcpp::wrap().

Here is the call graph for this function:



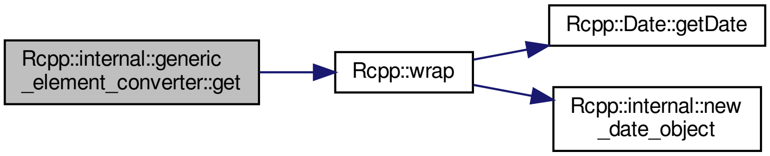
6.239.3.2 get() [2/3]

```
template<int RTYPE>
template<typename T >
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
              const T & input )  [inline], [static]
```

Definition at line 79 of file converter.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.239.3.3 get() [3/3]

```
template<int RTYPE>
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
              SEXP input )  [inline], [static]
```

Definition at line 87 of file converter.h.

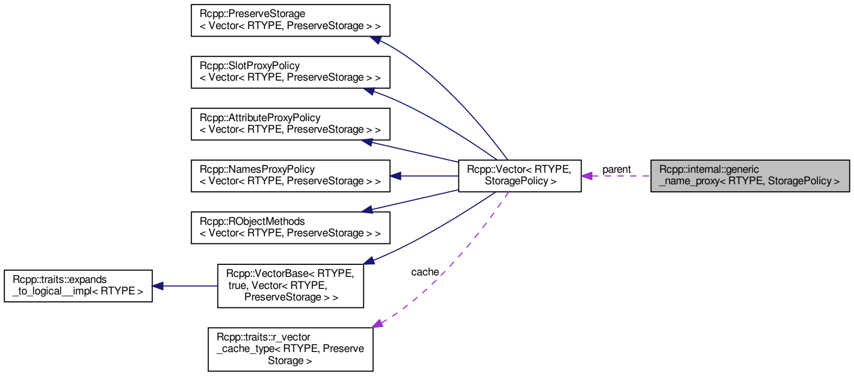
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.240 Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, StoragePolicy >
parent
Rcpp::internal::generic _name_proxy< RTYPE, StoragePolicy >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
cache
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR

Public Member Functions

- generic_name_proxy (VECTOR &v, const std::string &name_)
- generic_name_proxy (const generic_name_proxy &other)
- ~generic_name_proxy ()
- generic_name_proxy & operator= (SEXP rhs)
- generic_name_proxy & operator= (const generic_name_proxy &other)
- template< typename T >
  generic_name_proxy & operator= (const T &rhs)
- operator SEXP () const
- template< typename T >
  operator T () const

Private Member Functions

- void set (SEXP rhs)
- SEXP get () const

Private Attributes

- VECTOR & parent
- std::string name

6.240.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >

Definition at line 137 of file proxy.h.

6.240.2 Member Typedef Documentation

6.240.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::generic_name_proxy< RTYPE, Storage↩
Policy >::VECTOR
```

Definition at line 139 of file proxy.h.

6.240.3 Constructor & Destructor Documentation

6.240.3.1 generic_name_proxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::generic_name_proxy (
            VECTOR & v,
            const std::string & name_ )  [inline]
```

Definition at line 140 of file proxy.h.

6.240.3.2 generic_name_proxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::generic_name_proxy (
            const generic_name_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 142 of file proxy.h.

6.240.3.3 ∼generic_name_proxy()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::∼generic_name_proxy ( )  [inline]
```

Definition at line 144 of file proxy.h.

6.240.4 Member Function Documentation

6.240.4.1 get()

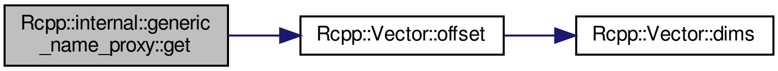
```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline], [private]
```

Definition at line 190 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator T(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



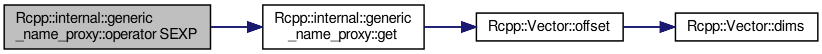
6.240.4.2 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

Definition at line 163 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



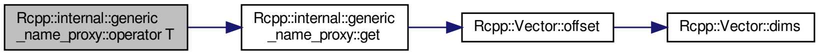
6.240.4.3 operator T()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator T ( ) const  [inline]
```

Definition at line 168 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get(), and RCPP_DEBUG_1.

Here is the call graph for this function:



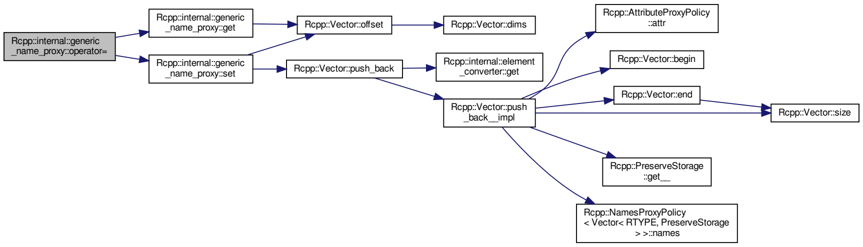
6.240.4.4 operator=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
             const generic_name_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 150 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



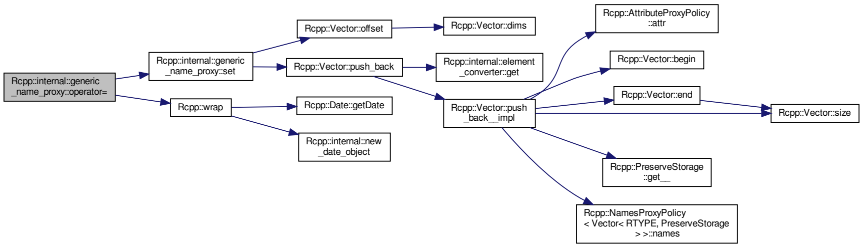
6.240.4.5 operator=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
            const T & rhs ) [inline]
```

Definition at line 156 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::wrap().

Here is the call graph for this function:


Rcpp::internal::generic_name_proxy::operator=
Rcpp::internal::generic_name_proxy::set
Rcpp::wrap
Rcpp::Vector::offset
Rcpp::Vector::push_back
Rcpp::Date::getDate
Rcpp::internal::new_date_object
Rcpp::Vector::dims
Rcpp::internal::element_converter::get
Rcpp::Vector::push_back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

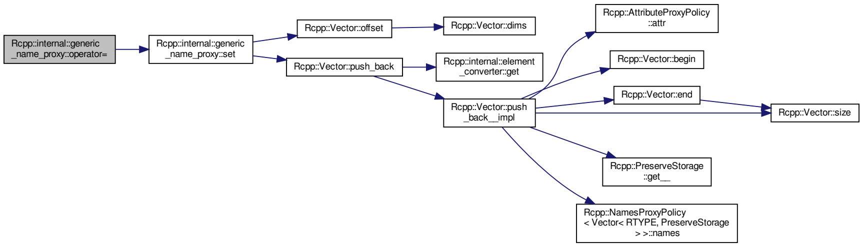
6.240.4.6 operator=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
            SEXP rhs ) [inline]
```

Definition at line 146 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::generic_name_proxy::operator=
Rcpp::internal::generic_name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::push_back
Rcpp::Vector::dims
Rcpp::internal::element_converter::get
Rcpp::Vector::push_back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

6.240.4.7 set()

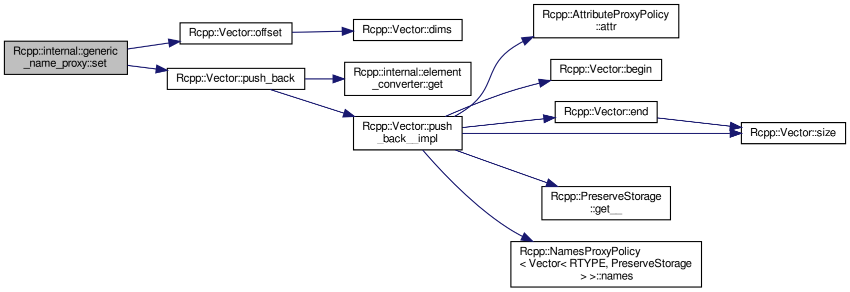
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set (
            SEXP rhs )  [inline], [private]
```

Definition at line 181 of file proxy.h.

References Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent, and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Referenced by Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:


Rcpp::internal::generic_name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::dims
Rcpp::Vector::push_back
Rcpp::internal::element_converter::get
Rcpp::Vector::push_back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

6.240.5 Member Data Documentation

6.240.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name  [private]
```

Definition at line 180 of file proxy.h.

Referenced by Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set().

6.240.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent  [private]
```

Definition at line 179 of file proxy.h.

Referenced by Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set().

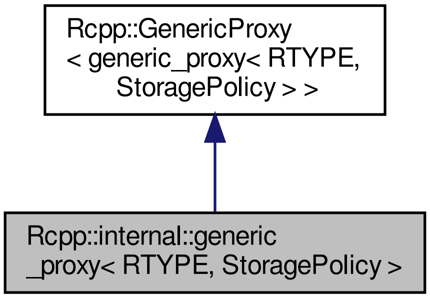
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

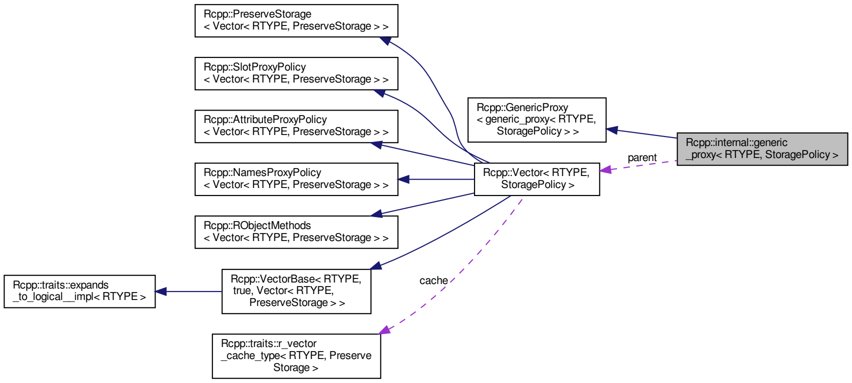
6.241 Rcpp::internal::generic_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <generic_proxy.h>
```

Inheritance diagram for Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >:



Collaboration diagram for Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >:



Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR

Public Member Functions

- generic_proxy ()
- generic_proxy (const generic_proxy &other)
- generic_proxy (VECTOR &v, R_xlen_t i)
- generic_proxy & operator= (SEXP rhs)
- generic_proxy & operator= (const generic_proxy &rhs)
- template<template< class > class StoragePolicy2>
  generic_proxy & operator= (const generic_proxy< RTYPE, StoragePolicy2 > &rhs)
- template<typename T >
  generic_proxy & operator= (const T &rhs)
- operator SEXP () const
- template<typename U >
  operator U () const
- operator bool () const
- operator int () const
- void swap (generic_proxy &other)
- void move (R_xlen_t n)
- void import (const generic_proxy &other)
- SEXP get () const

Public Attributes

- VECTOR ∗ parent
- R_xlen_t index

Private Member Functions

- void set (SEXP x)

6.241.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >

Definition at line 27 of file generic_proxy.h.

6.241.2 Member Typedef Documentation

6.241.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::generic_proxy< RTYPE, StoragePolicy
>::VECTOR
```

Definition at line 29 of file generic_proxy.h.

6.241.3 Constructor & Destructor Documentation

6.241.3.1 generic_proxy() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy ( )  [inline]
```

Definition at line 31 of file generic_proxy.h.

6.241.3.2 generic_proxy() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy (
            const generic_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 33 of file generic_proxy.h.

6.241.3.3 generic_proxy() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy (
            VECTOR & v,
            R_xlen_t i )  [inline]
```

Definition at line 37 of file generic_proxy.h.

6.241.4 Member Function Documentation

6.241.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline]
```

Definition at line 90 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator bool(), Rcpp::internal::generic_↩proxy< RTYPE, StoragePolicy >::operator int(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator U(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator=(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::swap().

6.241.4.2 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import (
            const generic_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 85 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::parent.

6.241.4.3 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::move (
            R_xlen_t n )  [inline]
```

Definition at line 83 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index.

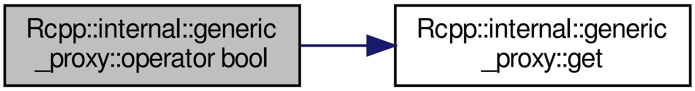
6.241.4.4 operator bool()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator bool ( ) const  [inline]
```

Definition at line 72 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



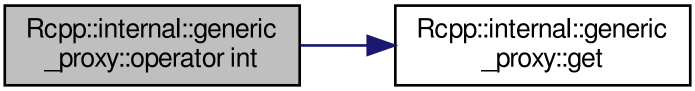
6.241.4.5 operator int()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator int ( ) const  [inline]
```

Definition at line 73 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:


Rcpp::internal::generic _proxy::operator int
Rcpp::internal::generic _proxy::get

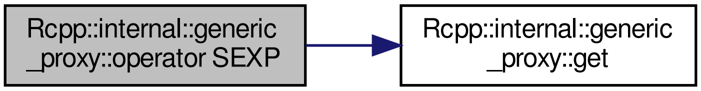
6.241.4.6 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

Definition at line 63 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:


Rcpp::internal::generic _proxy::operator SEXP
Rcpp::internal::generic _proxy::get

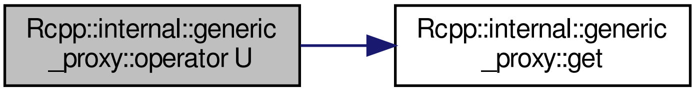
6.241.4.7 operator U()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename U >
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator U ( ) const  [inline]
```

Definition at line 67 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



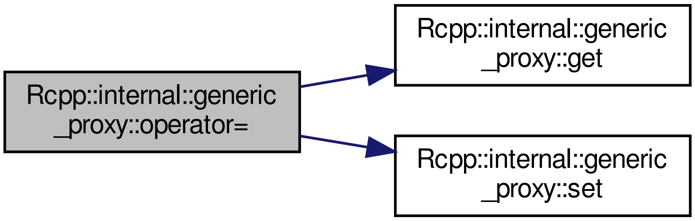
6.241.4.8 operator=() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
             const generic_proxy< RTYPE, StoragePolicy > & rhs )  [inline]
```

Definition at line 46 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



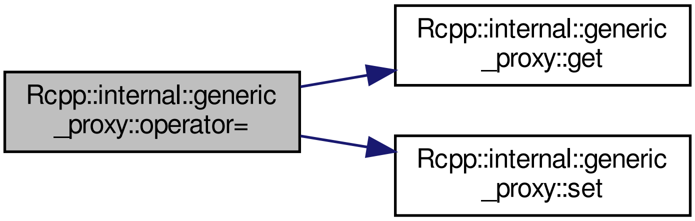
6.241.4.9 operator=() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
              const generic_proxy< RTYPE, StoragePolicy2 > & rhs )  [inline]
```

Definition at line 52 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



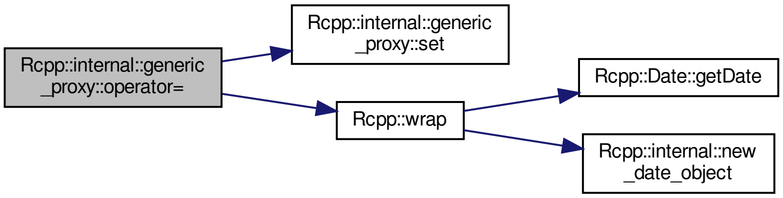
6.241.4.10 operator=() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
              const T & rhs )  [inline]
```

Definition at line 58 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::wrap().

Here is the call graph for this function:



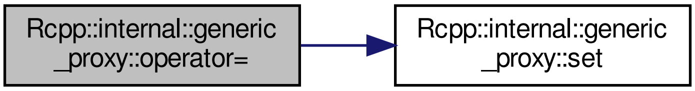
6.241.4.11 operator=() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
             SEXP rhs )  [inline]
```

Definition at line 41 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



6.241.4.12 set()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set (
             SEXP x )  [inline], [private]
```

Definition at line 95 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator=(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::swap().

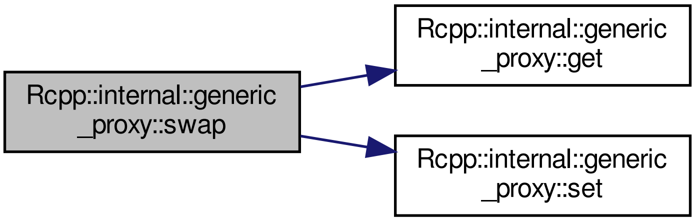
6.241.4.13 swap()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::swap (
            generic_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 75 of file generic_proxy.h.

References Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::generic_proxy::swap
Rcpp::internal::generic_proxy::get
Rcpp::internal::generic_proxy::set

6.241.5 Member Data Documentation

6.241.5.1 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 82 of file generic_proxy.h.

Referenced by Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::move(), and Rcpp↩ ::internal::generic_proxy< RTYPE, StoragePolicy >::set().

6.241.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR* Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 81 of file generic_proxy.h.

Referenced by Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import(), and Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/generic_proxy.h

6.242 Rcpp::GenericProxy< Proxy > Struct Template Reference

```
#include <GenericProxy.h>
```

Public Member Functions

- SEXP get () const

6.242.1 Detailed Description

template<typename Proxy>
struct Rcpp::GenericProxy< Proxy >

Definition at line 24 of file GenericProxy.h.

6.242.2 Member Function Documentation

6.242.2.1 get()

```
template<typename Proxy >
SEXP Rcpp::GenericProxy< Proxy >::get ( ) const  [inline]
```

Definition at line 25 of file GenericProxy.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

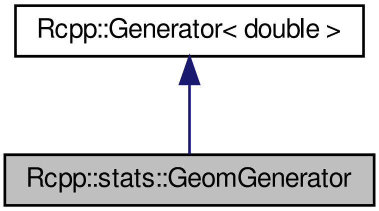
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/proxy/GenericProxy.h

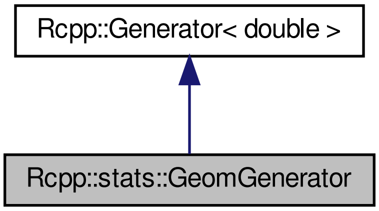
6.243 Rcpp::stats::GeomGenerator Class Reference

```
#include <rgeom.h>
```

Inheritance diagram for Rcpp::stats::GeomGenerator:



Collaboration diagram for Rcpp::stats::GeomGenerator:



Public Member Functions

- GeomGenerator (double p)
- double operator() () const

Private Attributes

- double lambda

Additional Inherited Members

6.243.1 Detailed Description

Definition at line 28 of file rgeom.h.

6.243.2 Constructor & Destructor Documentation

6.243.2.1 GeomGenerator()

```
Rcpp::stats::GeomGenerator::GeomGenerator (
            double p )  [inline]
```

Definition at line 31 of file rgeom.h.

6.243.3 Member Function Documentation

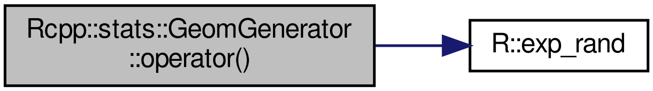
6.243.3.1 operator()()

```
double Rcpp::stats::GeomGenerator::operator() ( ) const  [inline]
```

Definition at line 33 of file rgeom.h.

References R::exp_rand(), and lambda.

Here is the call graph for this function:



6.243.4 Member Data Documentation

6.243.4.1 lambda

```
double Rcpp::stats::GeomGenerator::lambda  [private]
```

Definition at line 38 of file rgeom.h.

Referenced by operator()().

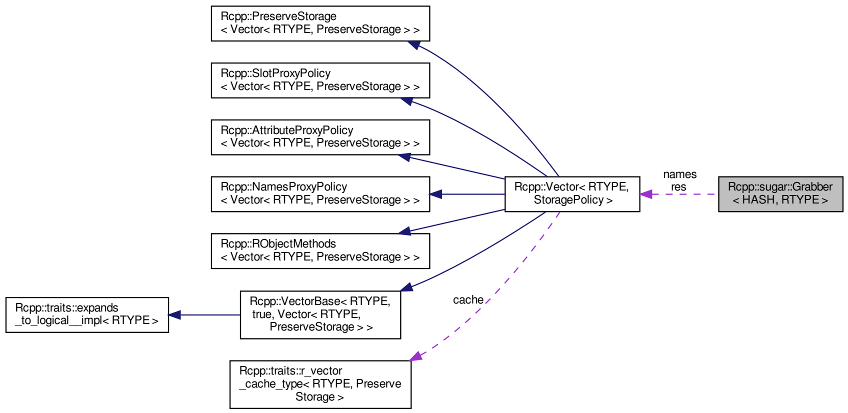
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rgeom.h

6.244 Rcpp::sugar::Grabber< HASH, RTYPE > Class Template Reference

```
#include <table.h>
```

Collaboration diagram for Rcpp::sugar::Grabber< HASH, RTYPE >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >
Rcpp::Vector< RTYPE, StoragePolicy >
cache
names
res
Rcpp::sugar::Grabber < HASH, RTYPE >

Public Member Functions

- Grabber (IntegerVector &res_, CharacterVector &names_)
- template<typename T >
  void operator() (T pair)

Private Attributes

- IntegerVector & res
- CharacterVector & names
- R_xlen_t index

6.244.1 Detailed Description

template<typename HASH, int RTYPE>
class Rcpp::sugar::Grabber< HASH, RTYPE >

Definition at line 42 of file table.h.

6.244.2 Constructor & Destructor Documentation

6.244.2.1 Grabber()

```
template<typename HASH , int RTYPE>
Rcpp::sugar::Grabber< HASH, RTYPE >::Grabber (
             IntegerVector & res_,
             CharacterVector & names_ )  [inline]
```

Definition at line 44 of file table.h.

6.244.3 Member Function Documentation

6.244.3.1 operator()()

```
template<typename HASH , int RTYPE>
template<typename T >
void Rcpp::sugar::Grabber< HASH, RTYPE >::operator() (
             T pair )  [inline]
```

Definition at line 47 of file table.h.

References Rcpp::sugar::Grabber< HASH, RTYPE >::index, Rcpp::sugar::Grabber< HASH, RTYPE >::names, and Rcpp::sugar::Grabber< HASH, RTYPE >::res.

6.244.4 Member Data Documentation

6.244.4.1 index

```
template<typename HASH , int RTYPE>
R_xlen_t Rcpp::sugar::Grabber< HASH, RTYPE >::index  [private]
```

Definition at line 55 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

6.244.4.2 names

```
template<typename HASH , int RTYPE>
CharacterVector& Rcpp::sugar::Grabber< HASH, RTYPE >::names  [private]
```

Definition at line 54 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

6.244.4.3 res

```
template<typename HASH , int RTYPE>
IntegerVector& Rcpp::sugar::Grabber< HASH, RTYPE >::res  [private]
```

Definition at line 53 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

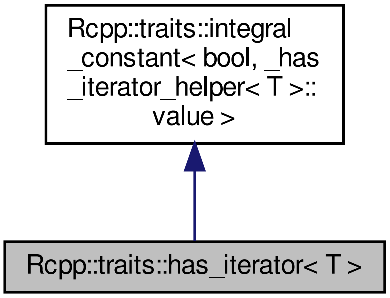
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

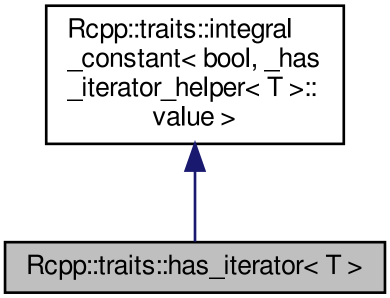
6.245 Rcpp::traits::has_iterator< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::has_iterator< T >:


Rcpp::traits::integral _constant< bool, _has _iterator_helper< T >:: value >
Rcpp::traits::has_iterator< T >

Collaboration diagram for Rcpp::traits::has_iterator< T >:


Rcpp::traits::integral _constant< bool, _has _iterator_helper< T >:: value >
Rcpp::traits::has_iterator< T >

Additional Inherited Members

6.245.1 Detailed Description

template<typename T>
struct Rcpp::traits::has_iterator< T >

uses the SFINAE idiom to check if a class has an nested iterator typedef. For example :

has_iterator< std::vector<int> >::value is true has_iterator< Rcpp::Symbol >::value is false

Definition at line 103 of file has_iterator.h.

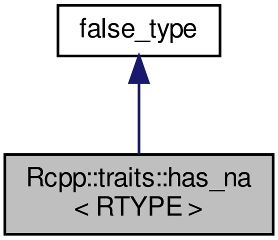
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

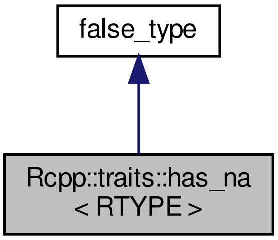
6.246 Rcpp::traits::has_na< RTYPE > Struct Template Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< RTYPE >:



Collaboration diagram for Rcpp::traits::has_na< RTYPE >:



Additional Inherited Members

6.246.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::has_na< RTYPE >

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.

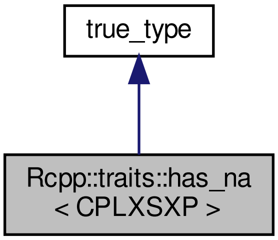
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

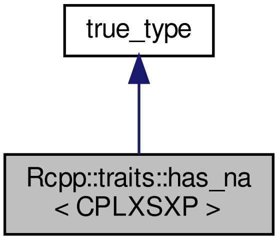
6.247 Rcpp::traits::has_na< CPLXSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< CPLXSXP >:



Collaboration diagram for Rcpp::traits::has_na< CPLXSXP >:



Additional Inherited Members

6.247.1 Detailed Description

complex vectors support missing values

Definition at line 50 of file has_na.h.

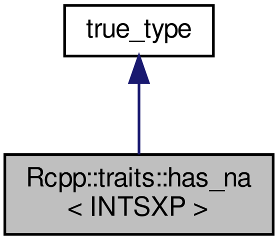
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

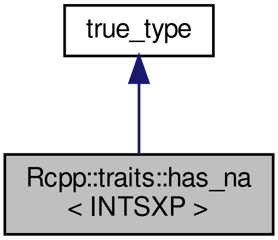
6.248 Rcpp::traits::has_na< INTSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< INTSXP >:



Collaboration diagram for Rcpp::traits::has_na< INTSXP >:



Additional Inherited Members

6.248.1 Detailed Description

integer vectors support missing values

Definition at line 40 of file has_na.h.

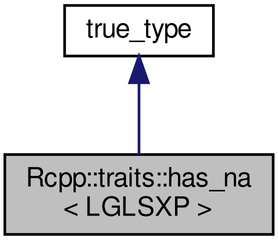
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

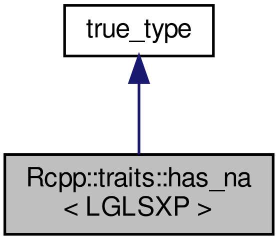
6.249 Rcpp::traits::has_na< LGLSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< LGLSXP >:



Collaboration diagram for Rcpp::traits::has_na< LGLSXP >:



Additional Inherited Members

6.249.1 Detailed Description

logical vectors support missing values

Definition at line 60 of file has_na.h.

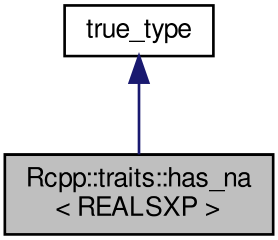
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

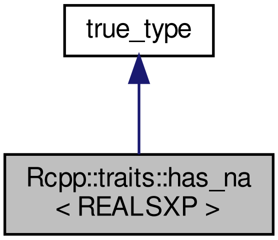
6.250 Rcpp::traits::has_na< REALSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< REALSXP >:



Collaboration diagram for Rcpp::traits::has_na< REALSXP >:



Additional Inherited Members

6.250.1 Detailed Description

numeric vectors support missing values

Definition at line 45 of file has_na.h.

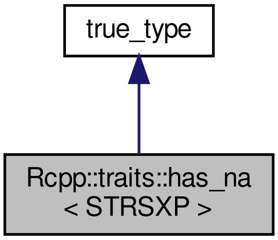
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

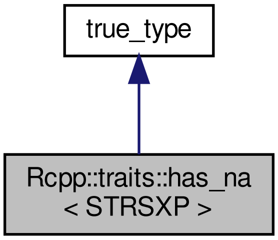
6.251 Rcpp::traits::has_na< STRSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has_na< STRSXP >:



Collaboration diagram for Rcpp::traits::has_na< STRSXP >:



Additional Inherited Members

6.251.1 Detailed Description

character vector support missing values

Definition at line 55 of file has_na.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_na.h

6.252 Rcpp::sugar::cbind_impl::detail::has_stored_type< T > Class Template Reference

```
#include <cbind.h>
```

Classes

- struct no

Static Public Attributes

- static const bool value = sizeof(test<T>(0)) == sizeof(yes)

Private Types

- typedef char yes

Static Private Member Functions

- template<typename C >
  static yes test (typename C::stored_type ∗)
- template<typename C >
  static no test (...)

6.252.1 Detailed Description

template<typename T>
class Rcpp::sugar::cbind_impl::detail::has_stored_type< T >

Definition at line 373 of file cbind.h.

6.252.2 Member Typedef Documentation

6.252.2.1 yes

```
template<typename T >
typedef char Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::yes  [private]
```

Definition at line 375 of file cbind.h.

6.252.3 Member Function Documentation

6.252.3.1 test() [1/2]

```
template<typename T >
template<typename C >
static no Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (
              ...  )  [static], [private]
```

6.252.3.2 test() [2/2]

```
template<typename T >
template<typename C >
static yes Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (
             typename C::stored_type *  )  [static], [private]
```

6.252.4 Member Data Documentation

6.252.4.1 value

```
template<typename T >
const bool Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::value = sizeof(test<T>(0)) ==
sizeof(yes)  [static]
```

Definition at line 387 of file cbind.h.

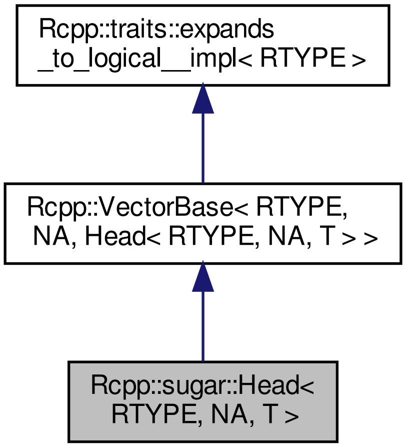
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

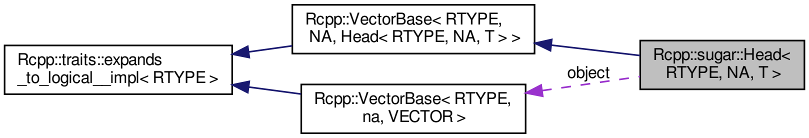
6.253 Rcpp::sugar::Head< RTYPE, NA, T > Class Template Reference

```
#include <head.h>
```

Inheritance diagram for Rcpp::sugar::Head< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Head< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Head (const VEC_TYPE &object_, R_xlen_t n_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- int n

6.253.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Head< RTYPE, NA, T >

Definition at line 29 of file head.h.

6.253.2 Member Typedef Documentation

6.253.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Head< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file head.h.

6.253.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Head< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file head.h.

6.253.3 Constructor & Destructor Documentation

6.253.3.1 Head()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Head< RTYPE, NA, T >::Head (
                 const VEC_TYPE & object_,
                 R_xlen_t n_ )  [inline]
```

Definition at line 34 of file head.h.

References Rcpp::sugar::Head< RTYPE, NA, T >::n.

6.253.4 Member Function Documentation

6.253.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Head< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 40 of file head.h.

6.253.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Head< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 43 of file head.h.

References Rcpp::sugar::Head< RTYPE, NA, T >::n.

6.253.5 Member Data Documentation

6.253.5.1 n

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Head< RTYPE, NA, T >::n  [private]
```

Definition at line 47 of file head.h.

Referenced by Rcpp::sugar::Head< RTYPE, NA, T >::Head(), and Rcpp::sugar::Head< RTYPE, NA, T >::size().

6.253.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Head< RTYPE, NA, T >::object  [private]
```

Definition at line 46 of file head.h.

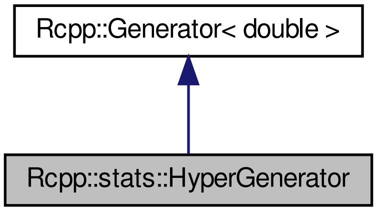
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/head.h

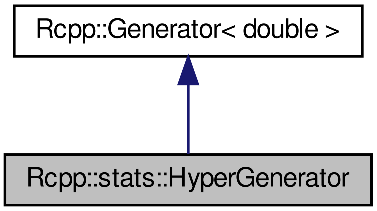
6.254 Rcpp::stats::HyperGenerator Class Reference

```
#include <rhyper.h>
```

Inheritance diagram for Rcpp::stats::HyperGenerator:



Collaboration diagram for Rcpp::stats::HyperGenerator:



Public Member Functions

- HyperGenerator (double nn1_, double nn2_, double kk_)
- double operator() () const

Private Attributes

- double nn1
- double nn2
- double kk

Additional Inherited Members

6.254.1 Detailed Description

Definition at line 28 of file rhyper.h.

6.254.2 Constructor & Destructor Documentation

6.254.2.1 HyperGenerator()

```
Rcpp::stats::HyperGenerator::HyperGenerator (
             double nn1_,
             double nn2_,
             double kk_ ) [inline]
```

Definition at line 30 of file rhyper.h.

6.254.3 Member Function Documentation

6.254.3.1 operator()()

```
double Rcpp::stats::HyperGenerator::operator() ( ) const [inline]
```

Definition at line 32 of file rhyper.h.

References kk, nn1, and nn2.

6.254.4 Member Data Documentation

6.254.4.1 kk

```
double Rcpp::stats::HyperGenerator::kk [private]
```

Definition at line 34 of file rhyper.h.

Referenced by operator()().

6.254.4.2 nn1

```
double Rcpp::stats::HyperGenerator::nn1  [private]
```

Definition at line 34 of file rhyper.h.

Referenced by operator()().

6.254.4.3 nn2

```
double Rcpp::stats::HyperGenerator::nn2  [private]
```

Definition at line 34 of file rhyper.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rhyper.h

6.255 Rcpp::traits::identity< T > Struct Template Reference

```
#include <traits.h>
```

Public Types

- typedef T type

6.255.1 Detailed Description

template<typename T>
struct Rcpp::traits::identity< T >

Definition at line 30 of file traits.h.

6.255.2 Member Typedef Documentation

6.255.2.1 type

```
template<typename T >
typedef T Rcpp::traits::identity< T >::type
```

Definition at line 30 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/traits.h

6.256 Rcpp::traits::if_< COND, LHS, RHS > Struct Template Reference

```
#include <if_.h>
```

Public Types

- typedef LHS type

6.256.1 Detailed Description

template<bool COND, typename LHS, typename RHS>
struct Rcpp::traits::if_< COND, LHS, RHS >

Definition at line 29 of file if_.h.

6.256.2 Member Typedef Documentation

6.256.2.1 type

```
template<bool COND, typename LHS , typename RHS >
typedef LHS Rcpp::traits::if_< COND, LHS, RHS >::type
```

Definition at line 30 of file if_.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/if_.h

6.257 Rcpp::traits::if_< false, LHS, RHS > Struct Template Reference

```
#include <if_.h>
```

Public Types

- typedef RHS type

6.257.1 Detailed Description

template<typename LHS, typename RHS>
struct Rcpp::traits::if_< false, LHS, RHS >

Definition at line 34 of file if_.h.

6.257.2 Member Typedef Documentation

6.257.2.1 type

```
template<typename LHS , typename RHS >
typedef RHS Rcpp::traits::if_< false, LHS, RHS >::type
```

Definition at line 35 of file if_.h.

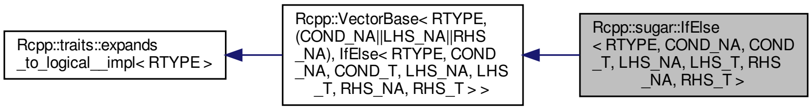
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/if_.h

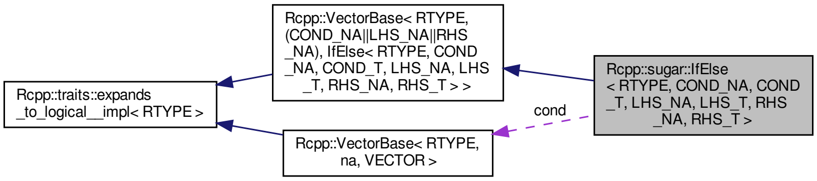
6.258 Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (COND_NA||LHS_NA||RHS _NA), IfElse< RTYPE, COND _NA, COND_T, LHS_NA, LHS _T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< RTYPE, (COND_NA||LHS_NA||RHS _NA), IfElse< RTYPE, COND _NA, COND_T, LHS_NA, LHS _T, RHS_NA, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::IfElse < RTYPE, COND_NA, COND _T, LHS_NA, LHS_T, RHS _NA, RHS_T >
cond
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,LHS_NA,LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE,RHS_NA,RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- IfElse (const COND_TYPE &cond_, const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- const LHS_T & lhs
- const RHS_T & rhs

6.258.1 Detailed Description

**template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >**

Definition at line 34 of file ifelse.h.

6.258.2 Member Typedef Documentation

6.258.2.1 COND_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T,
LHS_NA, LHS_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 40 of file ifelse.h.

6.258.2.2 LHS_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS↩
_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 41 of file ifelse.h.

6.258.2.3 RHS_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS↩
_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 42 of file ifelse.h.

6.258.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 43 of file ifelse.h.

6.258.3 Constructor & Destructor Documentation

6.258.3.1 IfElse()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse (
            const COND_TYPE & cond_,
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 48 of file ifelse.h.

References DEMANGLE, and RCPP_DEBUG.

6.258.4 Member Function Documentation

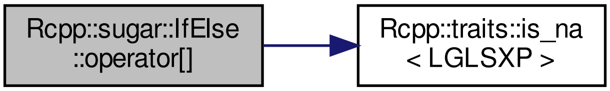
6.258.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
STORAGE Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 55 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp↩ ::traits::is_na< LGLSXP >(), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

Here is the call graph for this function:


Rcpp::sugar::IfElse ::operator[]
Rcpp::traits::is_na < LGLSXP >

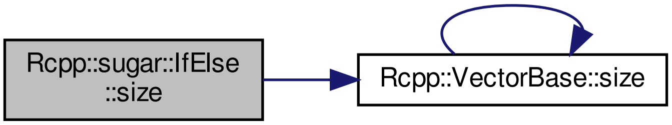
6.258.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 62 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.258.5 Member Data Documentation

6.258.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::cond [private]
```

Definition at line 65 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >↩ ::operator[ ](), Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), and Rcpp::sugar↩ ::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size().

6.258.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const LHS_T& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 66 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >↩ ::operator[ ](), and Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

6.258.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const RHS_T& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 67 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >↩ ::operator[ ](), and Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

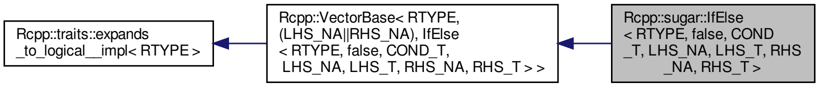
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

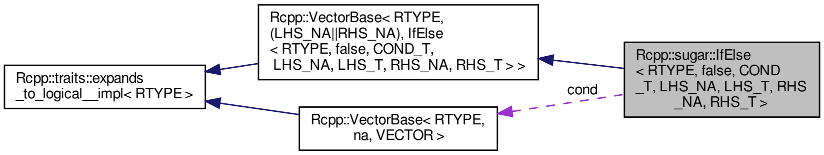
6.259 Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), IfElse < RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), IfElse < RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::IfElse < RTYPE, false, COND _T, LHS_NA, LHS_T, RHS _NA, RHS_T >
cond
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,LHS_NA,LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE,RHS_NA,RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE,LHS_NA,LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE,RHS_NA,RHS_T >::type RHS_EXT

Public Member Functions

- IfElse (const COND_TYPE &cond_, const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.259.1 Detailed Description

template<int RTYPE, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 77 of file ifelse.h.

6.259.2 Member Typedef Documentation

6.259.2.1 COND_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,COND_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 83 of file ifelse.h.

6.259.2.2 LHS_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse< RTYPE, false,
COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 88 of file ifelse.h.

6.259.2.3 LHS_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 84 of file ifelse.h.

6.259.2.4 RHS_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse< RTYPE, false,
COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 89 of file ifelse.h.

6.259.2.5 RHS_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 85 of file ifelse.h.

6.259.2.6 STORAGE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 86 of file ifelse.h.

6.259.3 Constructor & Destructor Documentation

6.259.3.1 IfElse()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse (
             const COND_TYPE & cond_,
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 91 of file ifelse.h.

6.259.4 Member Function Documentation

6.259.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 96 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

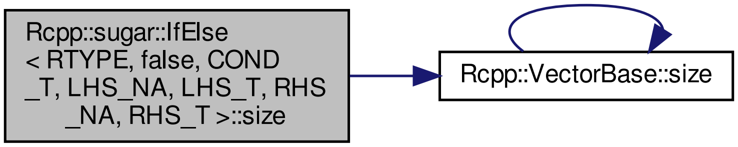
6.259.4.2 size()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 101 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:


Rcpp::sugar::IfElse < RTYPE, false, COND _T, LHS_NA, LHS_T, RHS _NA, RHS_T >::size
Rcpp::VectorBase::size

6.259.5 Member Data Documentation

6.259.5.1 cond

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond
[private]
```

Definition at line 105 of file ifelse.h.

6.259.5.2 lhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 106 of file ifelse.h.

6.259.5.3 rhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 107 of file ifelse.h.

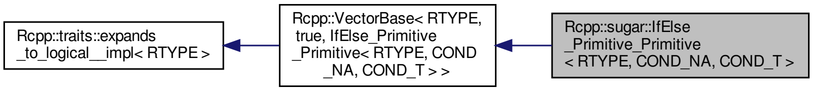
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

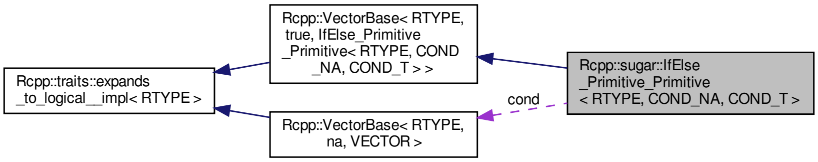
6.260 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >:



Collaboration diagram for Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >:



Public Types

- typedef Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > COND_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- IfElse_Primitive_Primitive (const COND_TYPE &cond_, STORAGE lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- STORAGE lhs
- STORAGE rhs
- STORAGE na

6.260.1 Detailed Description

template<int RTYPE, bool COND_NA, typename COND_T>
class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >

Definition at line 273 of file ifelse.h.

6.260.2 Member Typedef Documentation

6.260.2.1 COND_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE,
COND_NA, COND_T >::COND_TYPE
```

Definition at line 279 of file ifelse.h.

6.260.2.2 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_↩
NA, COND_T >::STORAGE
```

Definition at line 280 of file ifelse.h.

6.260.3 Constructor & Destructor Documentation

6.260.3.1 IfElse_Primitive_Primitive()

```
template<int RTYPE, bool COND_NA, typename COND_T >
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::IfElse_Primitive_Primitive (
            const COND_TYPE & cond_,
            STORAGE lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 282 of file ifelse.h.

6.260.4 Member Function Documentation

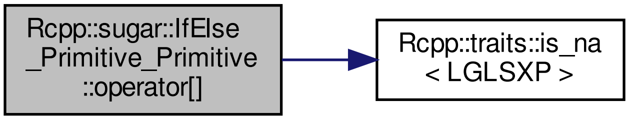
6.260.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 287 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond, Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::lhs, and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::rhs.

Here is the call graph for this function:



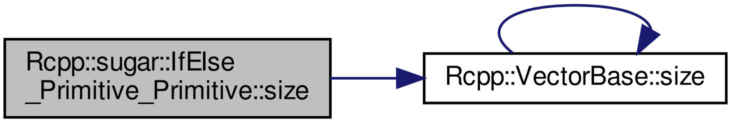
6.260.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::size ( ) const  [inline]
```

Definition at line 293 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.260.5 Member Data Documentation

6.260.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond  [private]
```

Definition at line 296 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[ ](), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[ ](), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::size(), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::size().

6.260.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::lhs  [private]
```

Definition at line 297 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[ ](), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[ ]().

6.260.5.3 na

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::na  [private]
```

Definition at line 299 of file ifelse.h.

6.260.5.4 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::rhs  [private]
```

Definition at line 298 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[ ](), and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[ ]().

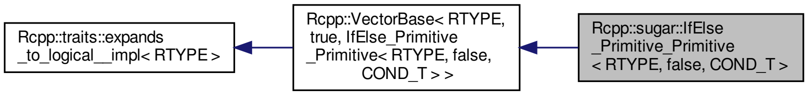
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

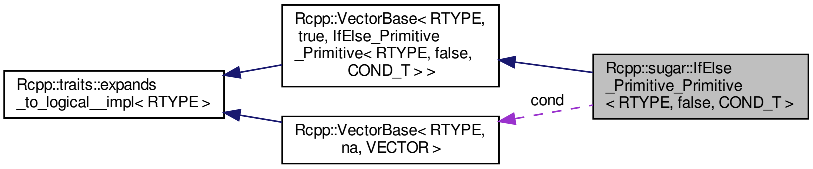
6.261 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >:



Collaboration diagram for Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >:



Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, COND_T > COND_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- IfElse_Primitive_Primitive (const COND_TYPE &cond_, STORAGE lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- STORAGE lhs
- STORAGE rhs

6.261.1 Detailed Description

template<int RTYPE, typename COND_T>
class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >

Definition at line 306 of file ifelse.h.

6.261.2 Member Typedef Documentation

6.261.2.1 COND_TYPE

```
template<int RTYPE, typename COND_T >
typedef Rcpp::VectorBase<LGLSXP,false,COND_T> Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE,
false, COND_T >::COND_TYPE
```

Definition at line 312 of file ifelse.h.

6.261.2.2 STORAGE

```
template<int RTYPE, typename COND_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false,
COND_T >::STORAGE
```

Definition at line 313 of file ifelse.h.

6.261.3 Constructor & Destructor Documentation

6.261.3.1 IfElse_Primitive_Primitive()

```
template<int RTYPE, typename COND_T >
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::IfElse_Primitive_Primitive (
             const COND_TYPE & cond_,
             STORAGE lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 315 of file ifelse.h.

6.261.4 Member Function Documentation

6.261.4.1 operator[]()

```
template<int RTYPE, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 320 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond, Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::lhs, and Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::rhs.

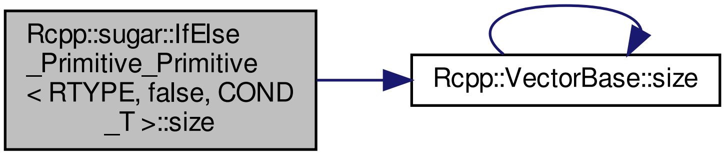
6.261.4.2 size()

```
template<int RTYPE, typename COND_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::size ( ) const  [inline]
```

Definition at line 324 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.261.5 Member Data Documentation

6.261.5.1 cond

```
template<int RTYPE, typename COND_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::cond  [private]
```

Definition at line 327 of file ifelse.h.

6.261.5.2 lhs

```
template<int RTYPE, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::lhs  [private]
```

Definition at line 328 of file ifelse.h.

6.261.5.3 rhs

```
template<int RTYPE, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::rhs  [private]
```

Definition at line 329 of file ifelse.h.

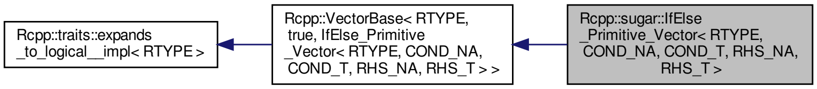
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

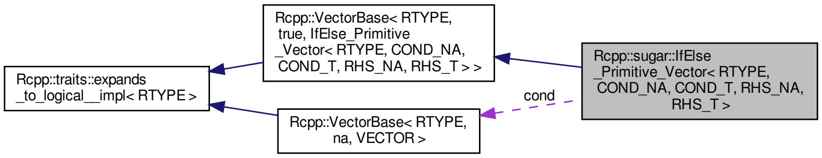
6.262 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive _Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< RTYPE, true, IfElse_Primitive _Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::IfElse _Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >
cond
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,RHS_NA,RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE,RHS_NA,RHS_T >::type RHS_EXT

Public Member Functions

- IfElse_Primitive_Vector (const COND_TYPE &cond_, STORAGE lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- STORAGE lhs
- const RHS_EXT & rhs

6.262.1 Detailed Description

**template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >**

Definition at line 119 of file ifelse.h.

6.262.2 Member Typedef Documentation

6.262.2.1 COND_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE,
COND_NA, COND_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 125 of file ifelse.h.

6.262.2.2 RHS_EXT

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse_Primitive_Vector<
RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 129 of file ifelse.h.

6.262.2.3 RHS_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND↩
_NA, COND_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 126 of file ifelse.h.

6.262.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA,
COND_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 127 of file ifelse.h.

6.262.3 Constructor & Destructor Documentation

6.262.3.1 IfElse_Primitive_Vector()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::IfElse_Primitive_Vector
(
             const COND_TYPE & cond_,
             STORAGE lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 131 of file ifelse.h.

6.262.4 Member Function Documentation

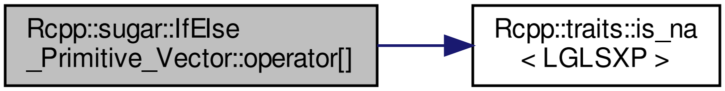
6.262.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 136 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond, Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs.

Here is the call graph for this function:



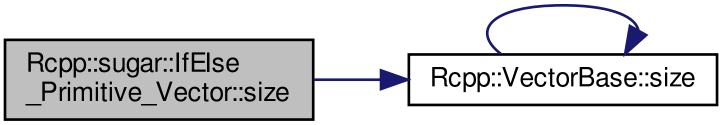
6.262.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 143 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.262.5 Member Data Documentation

6.262.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >↩
::cond  [private]
```

Definition at line 146 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar↩::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size(), and Rcpp::sugar::IfElse_↩Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size().

6.262.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 147 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[ ](), and Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[ ]().

6.262.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 148 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[ ](), and Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[ ]().

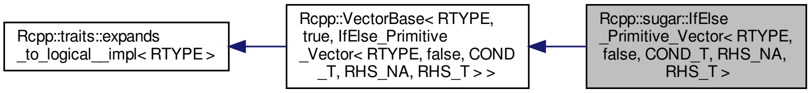
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

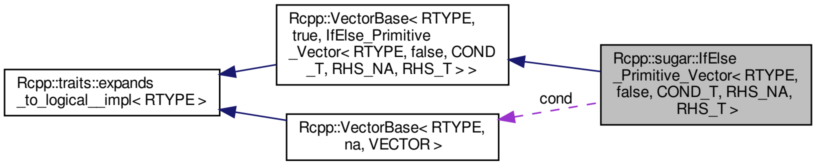
6.263 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,RHS_NA,RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE,RHS_NA,RHS_T >::type RHS_EXT

Public Member Functions

- IfElse_Primitive_Vector (const COND_TYPE &cond_, STORAGE lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- STORAGE lhs
- const RHS_EXT & rhs

6.263.1 Detailed Description

template<int RTYPE, typename COND_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >

Definition at line 157 of file ifelse.h.

6.263.2 Member Typedef Documentation

6.263.2.1 COND_TYPE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,COND_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 163 of file ifelse.h.

6.263.2.2 RHS_EXT

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse_Primitive_Vector<
RTYPE, false, COND_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 166 of file ifelse.h.

6.263.2.3 RHS_TYPE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 164 of file ifelse.h.

6.263.2.4 STORAGE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 165 of file ifelse.h.

6.263.3 Constructor & Destructor Documentation

6.263.3.1 IfElse_Primitive_Vector()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::IfElse_Primitive_Vector
(
             const COND_TYPE & cond_,
             STORAGE lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 168 of file ifelse.h.

6.263.4 Member Function Documentation

6.263.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 173 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond, Rcpp↩ ::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::IfElse_↩ Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs.

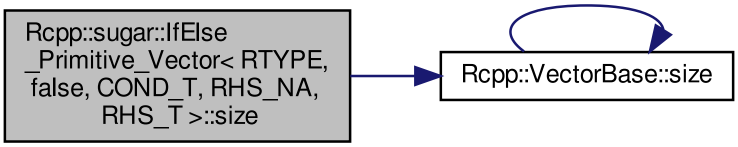
6.263.4.2 size()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 178 of file ifelse.h.

References Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.263.5 Member Data Documentation

6.263.5.1 cond

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >↩
::cond  [private]
```

Definition at line 181 of file ifelse.h.

6.263.5.2 lhs

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 182 of file ifelse.h.

6.263.5.3 rhs

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 183 of file ifelse.h.

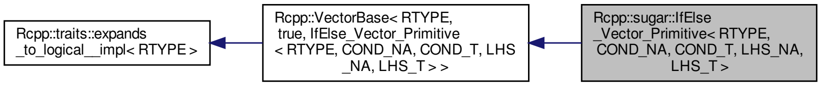
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

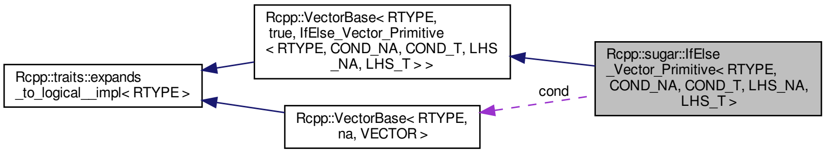
6.264 Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive < RTYPE, COND_NA, COND_T, LHS _NA, LHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive < RTYPE, COND_NA, COND_T, LHS _NA, LHS_T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Rcpp::sugar::IfElse _Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,LHS_NA,LHS_T > LHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE,LHS_NA,LHS_T >::type LHS_EXT

Public Member Functions

- IfElse_Vector_Primitive (const COND_TYPE &cond_, const LHS_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- const LHS_EXT & lhs
- const STORAGE rhs

6.264.1 Detailed Description

template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >

Definition at line 196 of file ifelse.h.

6.264.2 Member Typedef Documentation

6.264.2.1 COND_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE,
COND_NA, COND_T, LHS_NA, LHS_T >::COND_TYPE
```

Definition at line 202 of file ifelse.h.

6.264.2.2 LHS_EXT

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse_Vector_Primitive<
RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::LHS_EXT
```

Definition at line 205 of file ifelse.h.

6.264.2.3 LHS_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND↩
_NA, COND_T, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 203 of file ifelse.h.

6.264.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA,
COND_T, LHS_NA, LHS_T >::STORAGE
```

Definition at line 204 of file ifelse.h.

6.264.3 Constructor & Destructor Documentation

6.264.3.1 IfElse_Vector_Primitive()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::IfElse_Vector_Primitive
(
            const COND_TYPE & cond_,
            const LHS_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 207 of file ifelse.h.

6.264.4 Member Function Documentation

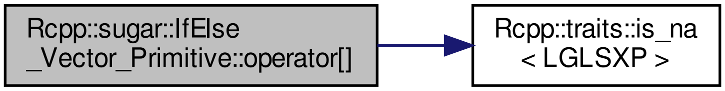
6.264.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 212 of file ifelse.h.

References Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond, Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::lhs, and Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::rhs.

Here is the call graph for this function:



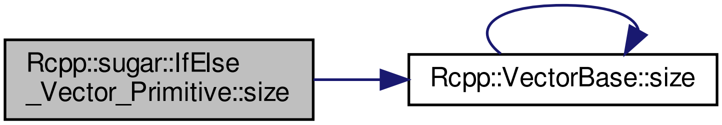
6.264.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::size ( )
const  [inline]
```

Definition at line 219 of file ifelse.h.

References Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.264.5 Member Data Documentation

6.264.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >↩
::cond  [private]
```

Definition at line 222 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::operator[ ](), Rcpp::sugar::If↩ Else_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::size(), and Rcpp::sugar::IfElse_Vector↩ _Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::size().

6.264.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const LHS_EXT& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::lhs
[private]
```

Definition at line 223 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[ ](), and Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::operator[ ]().

6.264.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::rhs
[private]
```

Definition at line 224 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[ ](), and Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::operator[ ]().

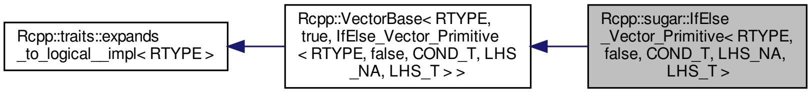
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

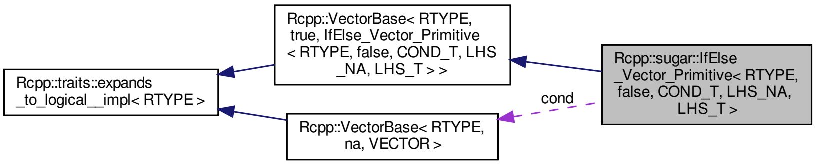
6.265 Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive < RTYPE, false, COND_T, LHS _NA, LHS_T > >
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 >:


Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive < RTYPE, false, COND_T, LHS _NA, LHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::IfElse _Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >
cond
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, COND_T > COND_TYPE
- typedef Rcpp::VectorBase< RTYPE,LHS_NA,LHS_T > LHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE,LHS_NA,LHS_T >::type LHS_EXT

Public Member Functions

- IfElse_Vector_Primitive (const COND_TYPE &cond_, const LHS_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const COND_TYPE & cond
- const LHS_EXT & lhs
- const STORAGE rhs

6.265.1 Detailed Description

template<int RTYPE, typename COND_T, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >

Definition at line 233 of file ifelse.h.

6.265.2 Member Typedef Documentation

6.265.2.1 COND_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,COND_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::COND_TYPE
```

Definition at line 239 of file ifelse.h.

6.265.2.2 LHS_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse_Vector_Primitive<
RTYPE, false, COND_T, LHS_NA, LHS_T >::LHS_EXT
```

Definition at line 242 of file ifelse.h.

6.265.2.3 LHS_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 240 of file ifelse.h.

6.265.2.4 STORAGE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::STORAGE
```

Definition at line 241 of file ifelse.h.

6.265.3 Constructor & Destructor Documentation

6.265.3.1 IfElse_Vector_Primitive()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::IfElse_Vector_Primitive
(
             const COND_TYPE & cond_,
             const LHS_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 244 of file ifelse.h.

6.265.4 Member Function Documentation

6.265.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 249 of file ifelse.h.

References Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond, Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::lhs, and Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::rhs.

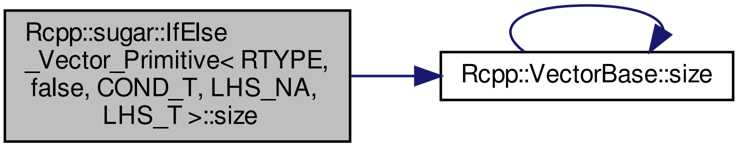
6.265.4.2 size()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::size ( )
const  [inline]
```

Definition at line 254 of file ifelse.h.

References Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.265.5 Member Data Documentation

6.265.5.1 cond

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >↩
::cond  [private]
```

Definition at line 257 of file ifelse.h.

6.265.5.2 lhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const LHS_EXT& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::lhs
[private]
```

Definition at line 258 of file ifelse.h.

6.265.5.3 rhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::rhs
[private]
```

Definition at line 259 of file ifelse.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

6.266 Rcpp::sugar::In< RTYPE, TABLE_T > Class Template Reference

```
#include <unique.h>
```

Public Member Functions

- In (const TABLE_T &table)
- template<typename T >
  LogicalVector get (const T &x) const

Private Types

- typedef sugar::IndexHash< RTYPE > HASH

Private Attributes

- Vector< RTYPE > vec
- HASH hash

6.266.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::In< RTYPE, TABLE_T >

Definition at line 44 of file unique.h.

6.266.2 Member Typedef Documentation

6.266.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef sugar::IndexHash<RTYPE> Rcpp::sugar::In< RTYPE, TABLE_T >::HASH  [private]
```

Definition at line 46 of file unique.h.

6.266.3 Constructor & Destructor Documentation

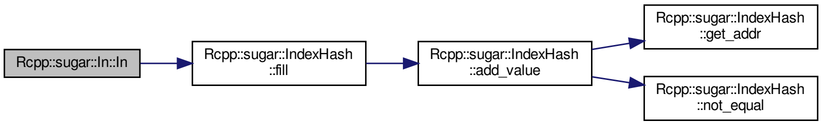
6.266.3.1 In()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::In< RTYPE, TABLE_T >::In (
             const TABLE_T & table )  [inline]
```

Definition at line 50 of file unique.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill(), and Rcpp::sugar::In< RTYPE, TABLE_T >::hash.

Here is the call graph for this function:


Rcpp::sugar::In::In
Rcpp::sugar::IndexHash ::fill
Rcpp::sugar::IndexHash ::add_value
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

6.266.4 Member Function Documentation

6.266.4.1 get()

```
template<int RTYPE, typename TABLE_T >
template<typename T >
LogicalVector Rcpp::sugar::In< RTYPE, TABLE_T >::get (
             const T & x ) const  [inline]
```

Definition at line 55 of file unique.h.

References Rcpp::sugar::In< RTYPE, TABLE_T >::hash.

Referenced by Rcpp::in().

6.266.5 Member Data Documentation

6.266.5.1 hash

```
template<int RTYPE, typename TABLE_T >
HASH Rcpp::sugar::In< RTYPE, TABLE_T >::hash  [private]
```

Definition at line 47 of file unique.h.

Referenced by Rcpp::sugar::In< RTYPE, TABLE_T >::get(), and Rcpp::sugar::In< RTYPE, TABLE_T >::In().

6.266.5.2 vec

```
template<int RTYPE, typename TABLE_T >
Vector<RTYPE> Rcpp::sugar::In< RTYPE, TABLE_T >::vec  [private]
```

Definition at line 45 of file unique.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/unique.h

6.267 Rcpp::sugar::IndexHash< RTYPE > Class Template Reference

```
#include <IndexHash.h>
```

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Vector< RTYPE > VECTOR

Public Member Functions

- IndexHash (SEXP table)
- IndexHash & fill ()
- LogicalVector fill_and_get_duplicated ()
- template<typename T >
  SEXP lookup (const T &vec) const
- SEXP lookup (const VECTOR &vec) const
- bool contains (STORAGE val) const
- int size () const
- Vector< RTYPE > keys () const
- template<typename T >
  SEXP lookup__impl (const T &vec, int n_) const
- SEXP get_profile_data ()
- bool not_equal (const STORAGE &lhs, const STORAGE &rhs)
- bool add_value (int i)
- uint32_t get_index (STORAGE value) const
- uint32_t get_addr (STORAGE value) const
- uint32_t get_addr (int value) const
- uint32_t get_addr (double val) const
- uint32_t get_addr (SEXP value) const

Public Attributes

- int n
- int m
- int k
- STORAGE ∗ src
- int size_
- int ∗ data

6.267.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::IndexHash< RTYPE >

Definition at line 49 of file IndexHash.h.

6.267.2 Member Typedef Documentation

6.267.2.1 STORAGE

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IndexHash< RTYPE >::STORAGE
```

Definition at line 51 of file IndexHash.h.

6.267.2.2 VECTOR

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::VECTOR
```

Definition at line 52 of file IndexHash.h.

6.267.3 Constructor & Destructor Documentation

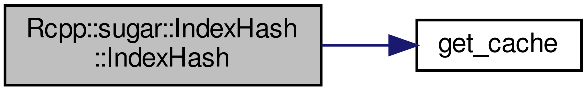
6.267.3.1 IndexHash()

```
template<int RTYPE>
Rcpp::sugar::IndexHash< RTYPE >::IndexHash (
            SEXP table ) [inline]
```

Definition at line 54 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, get_cache(), Rcpp::sugar::IndexHash< RTYPE >::k, Rcpp::sugar::IndexHash< RTYPE >::m, Rcpp::sugar::IndexHash< RTYPE >::n, RCPP_PROFILE_RECORD, RCPP_PROFILE_TIC, and RCPP_PROFILE_TOC.

Here is the call graph for this function:



6.267.4 Member Function Documentation

6.267.4.1 add_value()

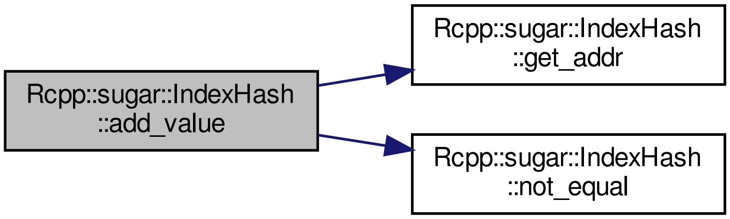
```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::add_value (
             int i )  [inline]
```

Definition at line 166 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, DEMANGLE, Rcpp::sugar::IndexHash< RTYPE >::get_addr(), Rcpp::sugar::IndexHash< RTYPE >::m, Rcpp::sugar::IndexHash< RTYPE >::not_equal(), RCPP_DEBUG_2, Rcpp↩ ::sugar::IndexHash< RTYPE >::size_, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), and Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_↩ duplicated().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::add_value
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

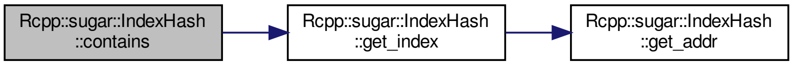
6.267.4.2 contains()

```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::contains (
             STORAGE val ) const  [inline]
```

Definition at line 101 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get_index().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::contains
Rcpp::sugar::IndexHash ::get_index
Rcpp::sugar::IndexHash ::get_addr

6.267.4.3 fill()

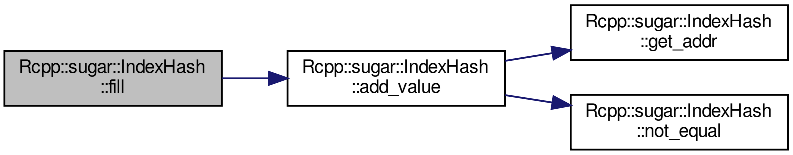
```
template<int RTYPE>
IndexHash& Rcpp::sugar::IndexHash< RTYPE >::fill ( )  [inline]
```

Definition at line 73 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::n, RCPP_PROFILE_RECORD, RCPP_PROFILE_TIC, and RCPP_PROFILE_TOC.

Referenced by Rcpp::sugar::In< RTYPE, TABLE_T >::In(), Rcpp::match(), and Rcpp::unique().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::fill
Rcpp::sugar::IndexHash ::add_value
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

6.267.4.4 fill_and_get_duplicated()

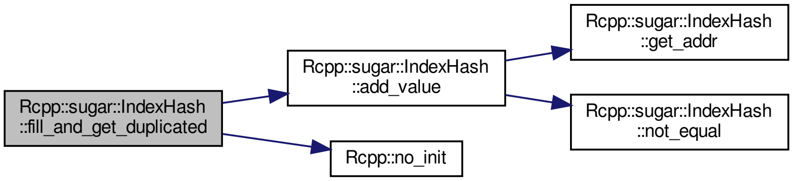
```
template<int RTYPE>
LogicalVector Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated ( )  [inline]
```

Definition at line 84 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::n, and Rcpp::no_init().

Referenced by Rcpp::duplicated().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::fill_and_get_duplicated
Rcpp::sugar::IndexHash ::add_value
Rcpp::no_init
Rcpp::sugar::IndexHash ::get_addr
Rcpp::sugar::IndexHash ::not_equal

6.267.4.5 get_addr() [1/4]

```
uint32_t Rcpp::sugar::IndexHash< REALSXP >::get_addr (
            double val ) const  [inline]
```

Definition at line 207 of file IndexHash.h.

References RCPP_HASH.

6.267.4.6 get_addr() [2/4]

```
uint32_t Rcpp::sugar::IndexHash< INTSXP >::get_addr (
            int value ) const  [inline]
```

Definition at line 203 of file IndexHash.h.

References RCPP_HASH.

6.267.4.7 get_addr() [3/4]

```
uint32_t Rcpp::sugar::IndexHash< STRSXP >::get_addr (
            SEXP value ) const  [inline]
```

Definition at line 224 of file IndexHash.h.

References RCPP_HASH.

6.267.4.8 get_addr() [4/4]

```
template<int RTYPE>
uint32_t Rcpp::sugar::IndexHash< RTYPE >::get_addr (
            STORAGE value ) const
```

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), and Rcpp::sugar::IndexHash< RTYPE >::get_index().

6.267.4.9 get_index()

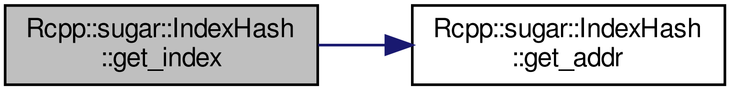
```
template<int RTYPE>
uint32_t Rcpp::sugar::IndexHash< RTYPE >::get_index (
             STORAGE value ) const  [inline]
```

Definition at line 187 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, Rcpp::sugar::IndexHash< RTYPE >::get_addr(), Rcpp::sugar::IndexHash< RTYPE >::m, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::contains(), and Rcpp::sugar::IndexHash< RTYPE >::lookup__impl().

Here is the call graph for this function:



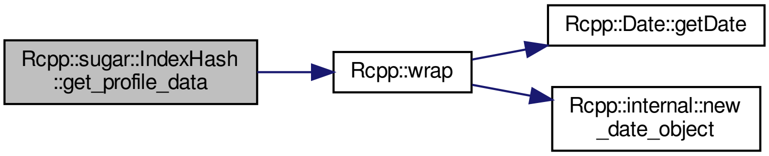
6.267.4.10 get_profile_data()

```
template<int RTYPE>
SEXP Rcpp::sugar::IndexHash< RTYPE >::get_profile_data ( )  [inline]
```

Definition at line 154 of file IndexHash.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.267.4.11 keys()

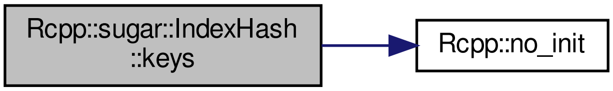
```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::keys ( ) const  [inline]
```

Definition at line 110 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, Rcpp::no_init(), Rcpp::sugar::IndexHash< RTYPE >::size_↩
, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::unique().

Here is the call graph for this function:



6.267.4.12 lookup() [1/2]

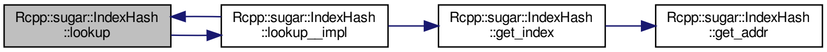
```
template<int RTYPE>
template<typename T >
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (
             const T & vec ) const  [inline]
```

Definition at line 92 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::lookup__impl().

Referenced by Rcpp::sugar::IndexHash< RTYPE >::lookup__impl(), and Rcpp::match().

Here is the call graph for this function:



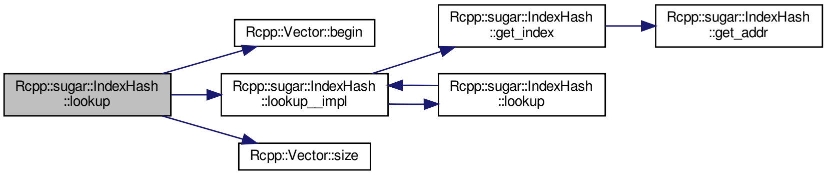
6.267.4.13 lookup() [2/2]

```
template<int RTYPE>
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (
            const VECTOR & vec ) const  [inline]
```

Definition at line 97 of file IndexHash.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::sugar::IndexHash< RTYPE >::lookup__impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::lookup
Rcpp::Vector::begin
Rcpp::sugar::IndexHash ::lookup__impl
Rcpp::Vector::size
Rcpp::sugar::IndexHash ::get_index
Rcpp::sugar::IndexHash ::lookup
Rcpp::sugar::IndexHash ::get_addr

6.267.4.14 lookup__impl()

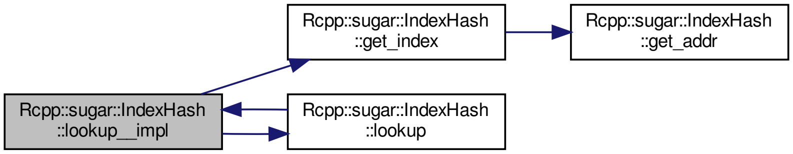
```
template<int RTYPE>
template<typename T >
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup__impl (
            const T & vec,
            int n_ ) const  [inline]
```

Definition at line 134 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get_index(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), RCPP_PROFILE_RECORD, RCPP_PROFILE_TIC, and RCPP_PROFILE_TOC.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::lookup().

Here is the call graph for this function:


Rcpp::sugar::IndexHash ::lookup__impl
Rcpp::sugar::IndexHash ::get_index
Rcpp::sugar::IndexHash ::lookup
Rcpp::sugar::IndexHash ::get_addr

6.267.4.15 not_equal()

```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::not_equal (
             const STORAGE & lhs,
             const STORAGE & rhs )  [inline]
```

Definition at line 162 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value().

6.267.4.16 size()

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::size ( ) const  [inline]
```

Definition at line 105 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::size_.

6.267.5 Member Data Documentation

6.267.5.1 data

```
template<int RTYPE>
int* Rcpp::sugar::IndexHash< RTYPE >::data
```

Definition at line 122 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::get_index(), Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

6.267.5.2 k

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::k
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

6.267.5.3 m

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::m
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::get_index(), and Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

6.267.5.4 n

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::n
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated(), and Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

6.267.5.5 size_

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::size_
```

Definition at line 120 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::keys(), and Rcpp::sugar::IndexHash< RTYPE >::size().

6.267.5.6 src

```
template<int RTYPE>
STORAGE* Rcpp::sugar::IndexHash< RTYPE >::src
```

Definition at line 119 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add_value(), Rcpp::sugar::IndexHash< RTYPE >::get_index(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/hash/IndexHash.h

6.268 Rcpp::traits::init_type< RTYPE > Struct Template Reference

```
#include <init_type.h>
```

Public Types

- typedef storage_type< RTYPE >::type type

6.268.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::init_type< RTYPE >

Definition at line 28 of file init_type.h.

6.268.2 Member Typedef Documentation

6.268.2.1 type

```
template<int RTYPE>
typedef storage_type<RTYPE>::type Rcpp::traits::init_type< RTYPE >::type
```

Definition at line 29 of file init_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/init_type.h

6.269 Rcpp::traits::init_type< LGLSXP > Struct Reference

```
#include <init_type.h>
```

Public Types

- typedef bool type

6.269.1 Detailed Description

Definition at line 34 of file init_type.h.

6.269.2 Member Typedef Documentation

6.269.2.1 type

```
typedef bool Rcpp::traits::init_type< LGLSXP >::type
```

Definition at line 35 of file init_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/init_type.h

6.270 Rcpp::traits::init_type< STRSXP > Struct Reference

```
#include <init_type.h>
```

Public Types

- typedef const char ∗ type

6.270.1 Detailed Description

Definition at line 31 of file init_type.h.

6.270.2 Member Typedef Documentation

6.270.2.1 type

```
typedef const char* Rcpp::traits::init_type< STRSXP >::type
```

Definition at line 32 of file init_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/init_type.h

6.271 Rcpp::traits::input_parameter< T > Struct Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef Rcpp::InputParameter< T > type

6.271.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< T >

Definition at line 82 of file InputParameter.h.

6.271.2 Member Typedef Documentation

6.271.2.1 type

```
template<typename T >
typedef Rcpp::InputParameter<T> Rcpp::traits::input_parameter< T >::type
```

Definition at line 83 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.272 Rcpp::traits::input_parameter< const T & > Struct Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef Rcpp::ConstReferenceInputParameter< T > type

6.272.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< const T & >

Definition at line 94 of file InputParameter.h.

6.272.2 Member Typedef Documentation

6.272.2.1 type

```
template<typename T >
typedef Rcpp::ConstReferenceInputParameter<T> Rcpp::traits::input_parameter< const T & >::type
```

Definition at line 95 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.273 Rcpp::traits::input_parameter< const T > Struct Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef Rcpp::ConstInputParameter< T > type

6.273.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< const T >

Definition at line 90 of file InputParameter.h.

6.273.2 Member Typedef Documentation

6.273.2.1 type

```
template<typename T >
typedef Rcpp::ConstInputParameter<T> Rcpp::traits::input_parameter< const T >::type
```

Definition at line 91 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.274 Rcpp::traits::input_parameter< T & > Struct Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef Rcpp::ReferenceInputParameter< T > type

6.274.1 Detailed Description

template<typename T>
struct Rcpp::traits::input_parameter< T & >

Definition at line 86 of file InputParameter.h.

6.274.2 Member Typedef Documentation

6.274.2.1 type

```
template<typename T >
typedef Rcpp::ReferenceInputParameter<T> Rcpp::traits::input_parameter< T & >::type
```

Definition at line 87 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.275 Rcpp::InputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

Public Member Functions

- InputParameter (SEXP x_)
- operator T ()

Private Attributes

- SEXP x

6.275.1 Detailed Description

template<typename T>
class Rcpp::InputParameter< T >

Definition at line 30 of file InputParameter.h.

6.275.2 Constructor & Destructor Documentation

6.275.2.1 InputParameter()

```
template<typename T >
Rcpp::InputParameter< T >::InputParameter (
             SEXP x_ )  [inline]
```

Definition at line 32 of file InputParameter.h.

6.275.3 Member Function Documentation

6.275.3.1 operator T()

```
template<typename T >
Rcpp::InputParameter< T >::operator T ( )  [inline]
```

Definition at line 34 of file InputParameter.h.

References Rcpp::InputParameter< T >::x.

6.275.4 Member Data Documentation

6.275.4.1 x

```
template<typename T >
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

6.276 Rcpp::sugar::InSet< HASH > Class Template Reference

```
#include <unique.h>
```

Public Member Functions

- InSet (const HASH &hash_)
- int operator() (STORAGE value)

Private Types

- typedef HASH::STORAGE STORAGE

Private Attributes

- const HASH & hash

6.276.1 Detailed Description

template<typename HASH>
class Rcpp::sugar::InSet< HASH >

Definition at line 29 of file unique.h.

6.276.2 Member Typedef Documentation

6.276.2.1 STORAGE

```
template<typename HASH >
typedef HASH::STORAGE Rcpp::sugar::InSet< HASH >::STORAGE  [private]
```

Definition at line 30 of file unique.h.

6.276.3 Constructor & Destructor Documentation

6.276.3.1 InSet()

```
template<typename HASH >
Rcpp::sugar::InSet< HASH >::InSet (
             const HASH & hash_ )  [inline]
```

Definition at line 33 of file unique.h.

6.276.4 Member Function Documentation

6.276.4.1 operator()()

```
template<typename HASH >
int Rcpp::sugar::InSet< HASH >::operator() (
             STORAGE value )  [inline]
```

Definition at line 35 of file unique.h.

References Rcpp::sugar::InSet< HASH >::hash.

6.276.5 Member Data Documentation

6.276.5.1 hash

```
template<typename HASH >
const HASH& Rcpp::sugar::InSet< HASH >::hash  [private]
```

Definition at line 40 of file unique.h.

Referenced by Rcpp::sugar::InSet< HASH >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/unique.h

6.277 Rcpp::traits::int2type< I > Struct Template Reference

```
#include <traits.h>
```

Public Types

- enum { value = I }

6.277.1 Detailed Description

template<int I>
struct Rcpp::traits::int2type< I >

Definition at line 33 of file traits.h.

6.277.2 Member Enumeration Documentation

6.277.2.1 anonymous enum

```
template<int I>
anonymous enum
```

Enumerator

value	

Definition at line 33 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/traits.h

6.278 Rcpp::traits::integral_constant< _T, _V > Struct Template Reference

```
#include <integral_constant.h>
```

Inherited by Rcpp::traits::same_type< r_type_traits< T >::r_category, r_type_primitive_tag >, Rcpp::traits::same_type< T::value_type, w
Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na, Rcpp::sugar::is_sugar_vector< T >, Rcpp::sugar::is_sugar_vector< Rcpp::Vecto
Rcpp::traits::has_na< RTYPE >, Rcpp::traits::has_na< CPLXSXP >, Rcpp::traits::has_na< INTSXP >, Rcpp::traits::has_na< LGLSXP
Rcpp::traits::has_na< REALSXP >, Rcpp::traits::has_na< STRSXP >, Rcpp::traits::is_arithmetic< typename >, Rcpp::traits::is_arithmetic< const double >, Rcpp::traits::is_arithmetic< const float >, Rcpp::traits::is_arithmetic< const int >, Rcpp::traits::is_arithmetic< const long >, Rcpp::traits::is_arithmetic< const long double >, Rcpp::traits::is_arithmetic< const short >, Rcpp::traits::is_arithmetic< const unsigned int >, Rcpp::traits::is_arithmetic< const unsigned long >, Rcpp::traits::is_arithmetic< const u
Rcpp::traits::is_arithmetic< double >, Rcpp::traits::is_arithmetic< float >, Rcpp::traits::is_arithmetic< int >, Rcpp::traits::is_arithmetic<
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< typenam
Rcpp::traits::is_const< _Tp const >, Rcpp::traits::is_named< T >, Rcpp::traits::is_named< Rcpp::Argument >, Rcpp::traits::is_named< named_object< T > >, Rcpp::traits::is_pointer< T >, Rcpp::traits::is_pointer< T * >, Rcpp::traits::is_reference< typename >, Rcpp::traits::is_reference< _Tp & >, Rcpp::traits::is_trivial< RTYPE >, Rcpp::traits::is_trivial< EXPRSXP >, Rcpp::traits::is_trivial< VECSXP >, Rcpp::traits::is_wide_string< char >, Rcpp::traits::is_wide_string< const char * >, Rcpp::traits::is_wide_string< const wchar_t * >, Rcpp::traits::is_wide_string< wchar_t >, Rcpp::traits::needs_protection< T >, Rcpp::traits::needs_protection< SEXP >, Rcpp::traits::r_sexptype_needscast< T >, Rcpp::traits::r_sexptype_needscast< Rbyte >, Rcpp::traits::r_sexptype_needscast< Rcomplex >, Rcpp::traits::r_sexptype_needscast<
Rcpp::traits::r_sexptype_needscast< int >, Rcpp::traits::same_type< T, U >, and Rcpp::traits::same_type< T, T >.

Public Types

- typedef _T value_type
- typedef integral_constant< _T, _V > type

Static Public Attributes

- static const _T value = _V

6.278.1 Detailed Description

template<typename _T, _T _V>
struct Rcpp::traits::integral_constant< _T, _V >

Definition at line 29 of file integral_constant.h.

6.278.2 Member Typedef Documentation

6.278.2.1 type

```
template<typename _T , _T _V>
typedef integral_constant<_T,_V> Rcpp::traits::integral_constant< _T, _V >::type
```

Definition at line 32 of file integral_constant.h.

6.278.2.2 value_type

```
template<typename _T , _T _V>
typedef _T Rcpp::traits::integral_constant< _T, _V >::value_type
```

Definition at line 31 of file integral_constant.h.

6.278.3 Member Data Documentation

6.278.3.1 value

```
template<typename _T , _T _V>
const _T Rcpp::traits::integral_constant< _T, _V >::value = _V  [static]
```

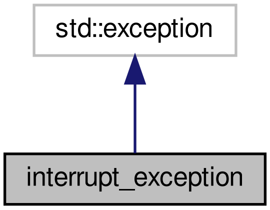
Definition at line 30 of file integral_constant.h.

The documentation for this struct was generated from the following file:

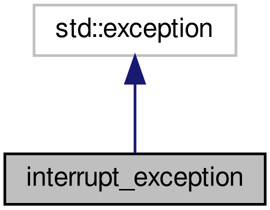
- inst/include/Rcpp/traits/integral_constant.h

6.279 interrupt_exception Class Reference

Inheritance diagram for interrupt_exception:



Collaboration diagram for interrupt_exception:



Public Member Functions

- interrupt_exception (std::string message)
- virtual ∼interrupt_exception () throw ()
- virtual const char ∗ what () const throw ()

Public Attributes

- std::string detailed_message

6.279.1 Detailed Description

Base class for interrupt exceptions thrown when user interrupts are detected.

Definition at line 15 of file piWithInterrupts.cpp.

6.279.2 Constructor & Destructor Documentation

6.279.2.1 interrupt_exception()

```
interrupt_exception::interrupt_exception (
            std::string message )  [inline]
```

Constructor.

Parameters

in	*message*	A description of event that caused this exception.

Definition at line 22 of file piWithInterrupts.cpp.

6.279.2.2 ∼interrupt_exception()

```
virtual interrupt_exception::∼interrupt_exception ( ) throw ( )   [inline], [virtual]
```

Virtual destructor. Needed to avoid "looser throw specification" errors.

Definition at line 29 of file piWithInterrupts.cpp.

6.279.3 Member Function Documentation

6.279.3.1 what()

```
virtual const char∗ interrupt_exception::what ( ) const throw ( )   [inline], [virtual]
```

Obtain a description of the exception.

Returns

Description.

Definition at line 35 of file piWithInterrupts.cpp.

References detailed_message.

6.279.4 Member Data Documentation

6.279.4.1 detailed_message

```
std::string interrupt_exception::detailed_message
```

String with details on the error.

Definition at line 42 of file piWithInterrupts.cpp.

Referenced by what().

The documentation for this class was generated from the following file:

- inst/examples/OpenMP/piWithInterrupts.cpp

6.280 Rcpp::internal::InterruptedException Class Reference

```
#include <Interrupt.h>
```

6.280.1 Detailed Description

Definition at line 33 of file Interrupt.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Interrupt.h

6.281 Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <setdiff.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Intersect (const LHS_T &lhs, const RHS_T &rhs)
- Vector< RTYPE > get () const

Private Types

- typedef RCPP_UNORDERED_SET< STORAGE > SET
- typedef SET::const_iterator ITERATOR

Private Attributes

- SET intersect

6.281.1 Detailed Description

**template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >**

Definition at line 100 of file setdiff.h.

6.281.2 Member Typedef Documentation

6.281.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 127 of file setdiff.h.

6.281.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T
>::SET  [private]
```

Definition at line 126 of file setdiff.h.

6.281.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 102 of file setdiff.h.

6.281.3 Constructor & Destructor Documentation

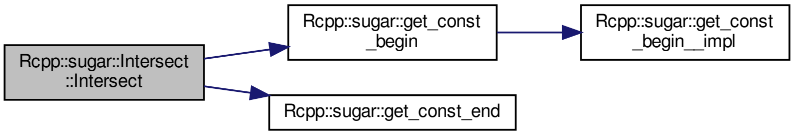
6.281.3.1 Intersect()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect (
            const LHS_T & lhs,
            const RHS_T & rhs )  [inline]
```

Definition at line 104 of file setdiff.h.

References Rcpp::sugar::get_const_begin(), Rcpp::sugar::get_const_end(), and Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::intersect.

Here is the call graph for this function:


Rcpp::sugar::Intersect ::Intersect
Rcpp::sugar::get_const _begin
Rcpp::sugar::get_const _begin__impl
Rcpp::sugar::get_const_end

6.281.4 Member Function Documentation

6.281.4.1 get()

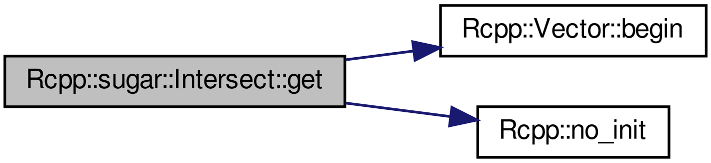
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const  [inline]
```

Definition at line 118 of file setdiff.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::intersect, and Rcpp::no_init().

Referenced by Rcpp::intersect().

Here is the call graph for this function:



6.281.5 Member Data Documentation

6.281.5.1 intersect

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::intersect  [private]
```

Definition at line 128 of file setdiff.h.

Referenced by Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), and Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect().

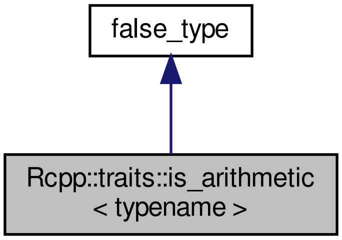
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

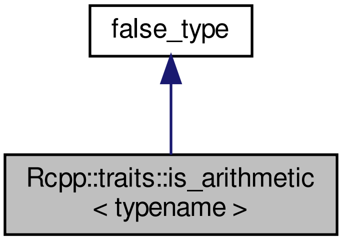
6.282 Rcpp::traits::is_arithmetic< typename > Struct Template Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< typename >:



Collaboration diagram for Rcpp::traits::is_arithmetic< typename >:



Additional Inherited Members

6.282.1 Detailed Description

template<typename>
struct Rcpp::traits::is_arithmetic< typename >

Definition at line 30 of file is_arithmetic.h.

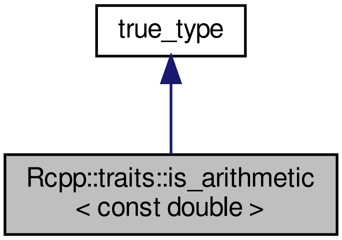
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

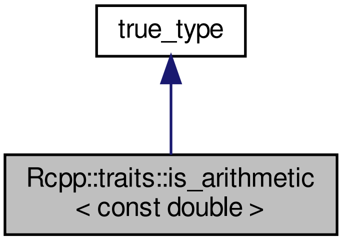
6.283 Rcpp::traits::is_arithmetic< const double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const double >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const double >:



Additional Inherited Members

6.283.1 Detailed Description

Definition at line 94 of file is_arithmetic.h.

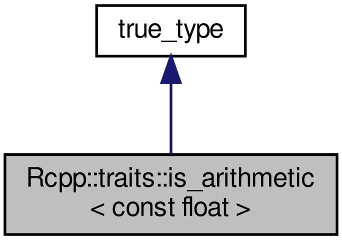
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

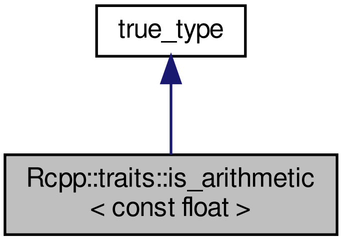
6.284 Rcpp::traits::is_arithmetic< const float > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const float >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const float >:



Additional Inherited Members

6.284.1 Detailed Description

Definition at line 88 of file is_arithmetic.h.

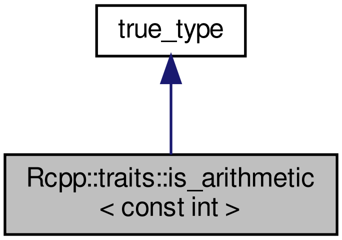
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

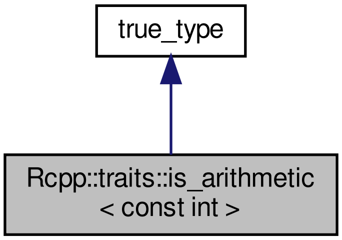
6.285 Rcpp::traits::is_arithmetic< const int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const int >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const int >:



Additional Inherited Members

6.285.1 Detailed Description

Definition at line 48 of file is_arithmetic.h.

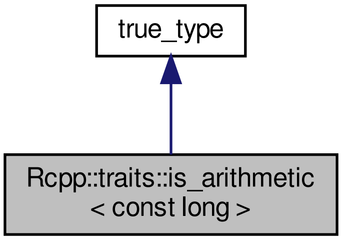
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

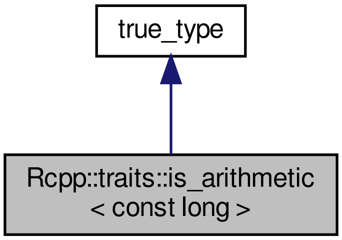
6.286 Rcpp::traits::is_arithmetic< const long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const long >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const long >:



Additional Inherited Members

6.286.1 Detailed Description

Definition at line 60 of file is_arithmetic.h.

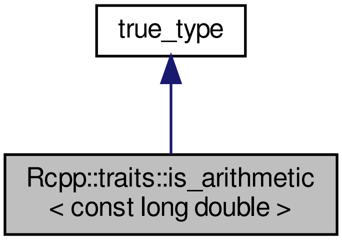
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

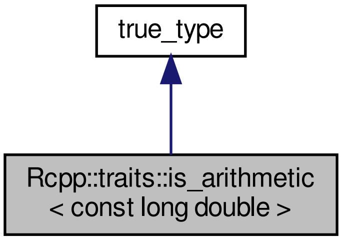
6.287 Rcpp::traits::is_arithmetic< const long double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const long double >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const long double >:



Additional Inherited Members

6.287.1 Detailed Description

Definition at line 100 of file is_arithmetic.h.

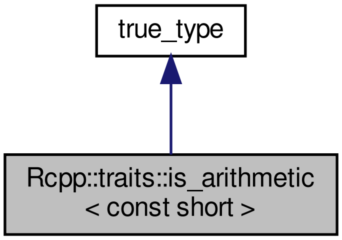
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

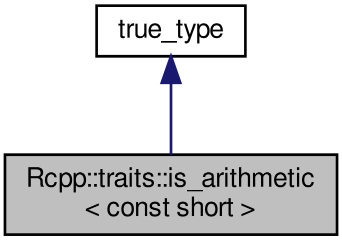
6.288 Rcpp::traits::is_arithmetic< const short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const short >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const short >:



Additional Inherited Members

6.288.1 Detailed Description

Definition at line 36 of file is_arithmetic.h.

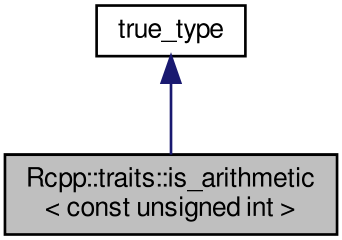
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

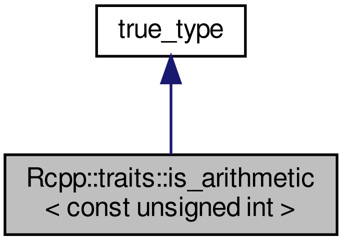
6.289 Rcpp::traits::is_arithmetic< const unsigned int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const unsigned int >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const unsigned int >:



Additional Inherited Members

6.289.1 Detailed Description

Definition at line 54 of file is_arithmetic.h.

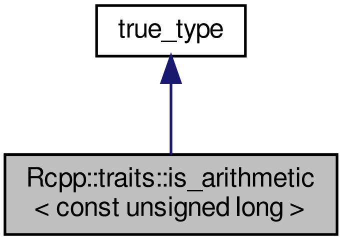
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

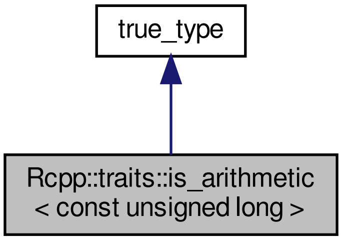
6.290 Rcpp::traits::is_arithmetic< const unsigned long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const unsigned long >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const unsigned long >:



Additional Inherited Members

6.290.1 Detailed Description

Definition at line 66 of file is_arithmetic.h.

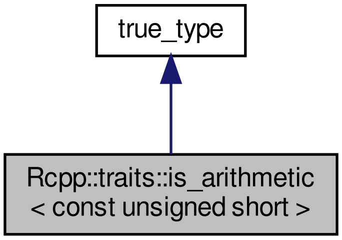
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

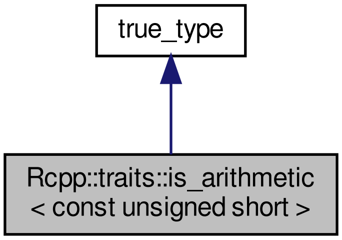
6.291 Rcpp::traits::is_arithmetic< const unsigned short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< const unsigned short >:



Collaboration diagram for Rcpp::traits::is_arithmetic< const unsigned short >:



Additional Inherited Members

6.291.1 Detailed Description

Definition at line 42 of file is_arithmetic.h.

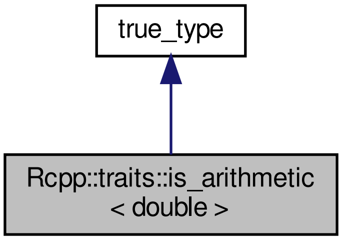
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

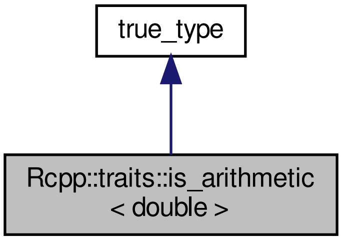
6.292 Rcpp::traits::is_arithmetic< double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< double >:



Collaboration diagram for Rcpp::traits::is_arithmetic< double >:



Additional Inherited Members

6.292.1 Detailed Description

Definition at line 91 of file is_arithmetic.h.

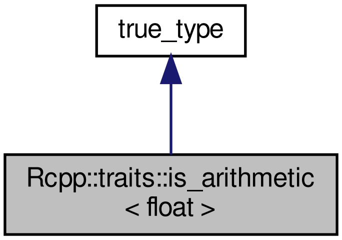
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

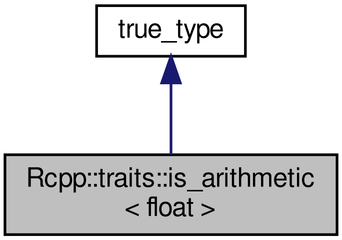
6.293 Rcpp::traits::is_arithmetic< float > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< float >:



Collaboration diagram for Rcpp::traits::is_arithmetic< float >:



Additional Inherited Members

6.293.1 Detailed Description

Definition at line 85 of file is_arithmetic.h.

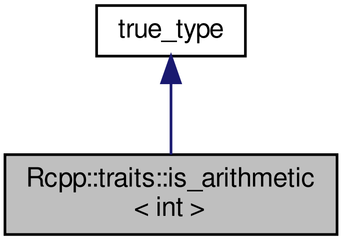
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

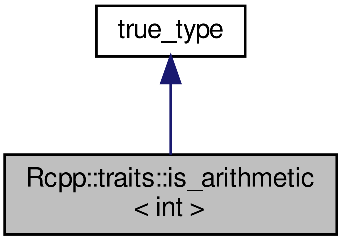
6.294 Rcpp::traits::is_arithmetic< int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< int >:



Collaboration diagram for Rcpp::traits::is_arithmetic< int >:



Additional Inherited Members

6.294.1 Detailed Description

Definition at line 45 of file is_arithmetic.h.

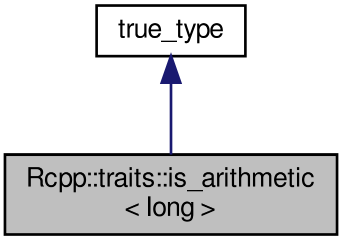
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

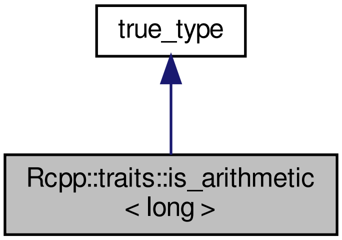
6.295 Rcpp::traits::is_arithmetic< long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< long >:



Collaboration diagram for Rcpp::traits::is_arithmetic< long >:



Additional Inherited Members

6.295.1 Detailed Description

Definition at line 57 of file is_arithmetic.h.

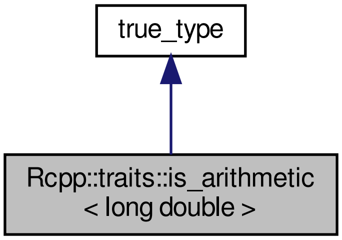
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

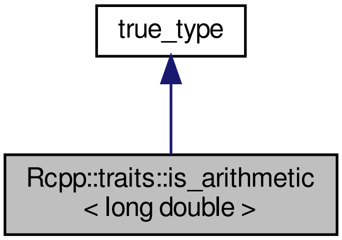
6.296 Rcpp::traits::is_arithmetic< long double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< long double >:



Collaboration diagram for Rcpp::traits::is_arithmetic< long double >:



Additional Inherited Members

6.296.1 Detailed Description

Definition at line 97 of file is_arithmetic.h.

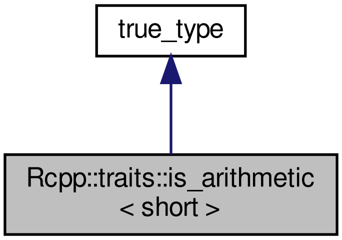
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

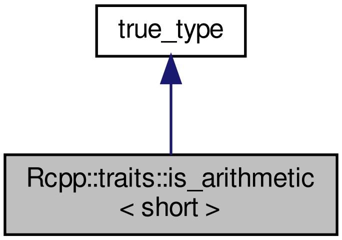
6.297 Rcpp::traits::is_arithmetic< short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< short >:



Collaboration diagram for Rcpp::traits::is_arithmetic< short >:



Additional Inherited Members

6.297.1 Detailed Description

Definition at line 33 of file is_arithmetic.h.

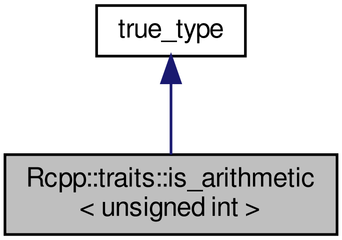
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

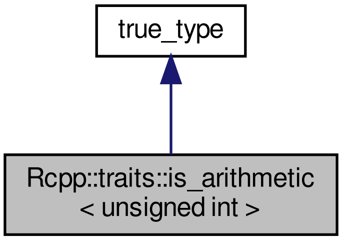
6.298 Rcpp::traits::is_arithmetic< unsigned int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< unsigned int >:



Collaboration diagram for Rcpp::traits::is_arithmetic< unsigned int >:



Additional Inherited Members

6.298.1 Detailed Description

Definition at line 51 of file is_arithmetic.h.

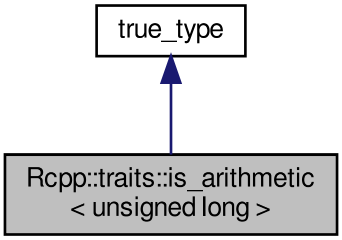
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

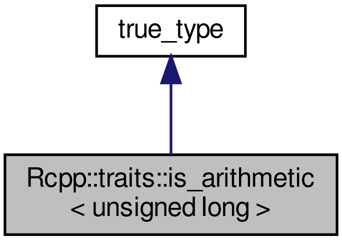
6.299 Rcpp::traits::is_arithmetic< unsigned long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< unsigned long >:



Collaboration diagram for Rcpp::traits::is_arithmetic< unsigned long >:



Additional Inherited Members

6.299.1 Detailed Description

Definition at line 63 of file is_arithmetic.h.

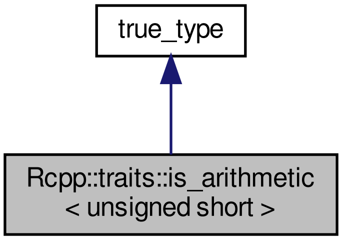
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

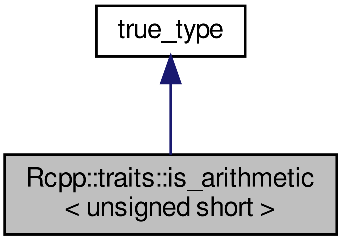
6.300 Rcpp::traits::is_arithmetic< unsigned short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is_arithmetic< unsigned short >:



Collaboration diagram for Rcpp::traits::is_arithmetic< unsigned short >:



Additional Inherited Members

6.300.1 Detailed Description

Definition at line 39 of file is_arithmetic.h.

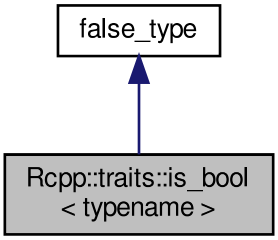
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_arithmetic.h

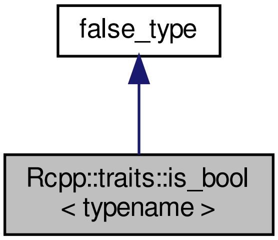
6.301 Rcpp::traits::is_bool< typename > Struct Template Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is_bool< typename >:



Collaboration diagram for Rcpp::traits::is_bool< typename >:



Additional Inherited Members

6.301.1 Detailed Description

template<typename>
struct Rcpp::traits::is_bool< typename >

Definition at line 7 of file is_bool.h.

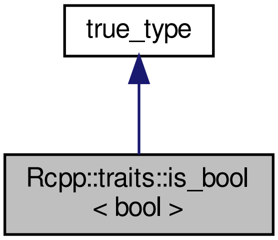
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

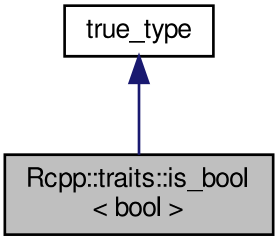
6.302 Rcpp::traits::is_bool< bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is_bool< bool >:



Collaboration diagram for Rcpp::traits::is_bool< bool >:



Additional Inherited Members

6.302.1 Detailed Description

Definition at line 11 of file is_bool.h.

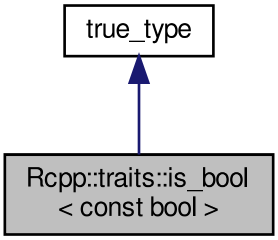
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

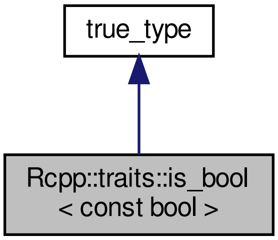
6.303 Rcpp::traits::is_bool< const bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is_bool< const bool >:



Collaboration diagram for Rcpp::traits::is_bool< const bool >:



Additional Inherited Members

6.303.1 Detailed Description

Definition at line 15 of file is_bool.h.

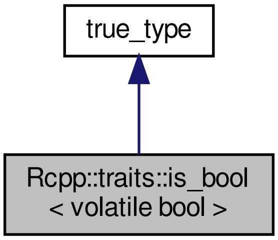
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_bool.h

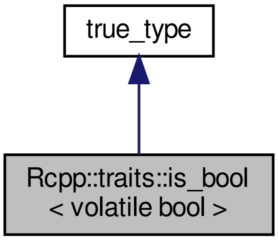
6.304 Rcpp::traits::is_bool< volatile bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is_bool< volatile bool >:



Collaboration diagram for Rcpp::traits::is_bool< volatile bool >:



Additional Inherited Members

6.304.1 Detailed Description

Definition at line 19 of file is_bool.h.

The documentation for this struct was generated from the following file:

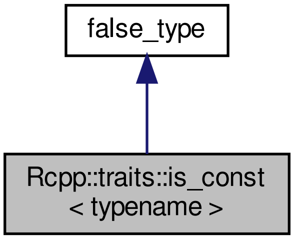
- inst/include/Rcpp/traits/is_bool.h

6.305 Rcpp::traits::is_const< typename > Struct Template Reference

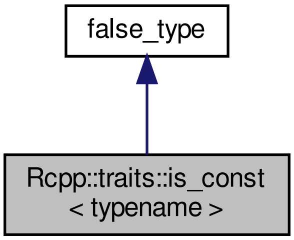
type properties [4.5.3].

```
#include <is_const.h>
```

Inheritance diagram for Rcpp::traits::is_const< typename >:



Collaboration diagram for Rcpp::traits::is_const< typename >:



Additional Inherited Members

6.305.1 Detailed Description

template<typename>
struct Rcpp::traits::is_const< typename >

type properties [4.5.3].

Definition at line 30 of file is_const.h.

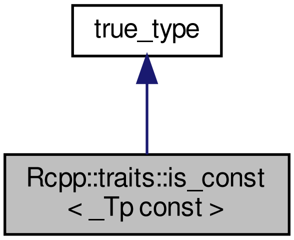
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_const.h

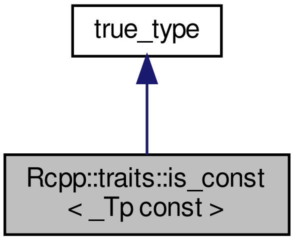
6.306 Rcpp::traits::is_const< _Tp const > Struct Template Reference

```
#include <is_const.h>
```

Inheritance diagram for Rcpp::traits::is_const< _Tp const >:



Collaboration diagram for Rcpp::traits::is_const< _Tp const >:



Additional Inherited Members

6.306.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::is_const< _Tp const >

Definition at line 34 of file is_const.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_const.h

6.307 Rcpp::traits::is_convertible< T, U > Class Template Reference

```
#include <is_convertible.h>
```

Classes

- struct Big

Public Types

- typedef bool value_type
- typedef integral_constant< bool, value > type

Static Public Attributes

- static const bool value = sizeof(Test(MakeT())) == sizeof(Small)

Private Types

- typedef char Small

Static Private Member Functions

- static Small Test (const U &)
- static Big Test (...)
- static T MakeT ()

6.307.1 Detailed Description

template<typename T, typename U>
class Rcpp::traits::is_convertible< T, U >

Definition at line 29 of file is_convertible.h.

6.307.2 Member Typedef Documentation

6.307.2.1 Small

```
template<typename T , typename U >
typedef char Rcpp::traits::is_convertible< T, U >::Small  [private]
```

Definition at line 31 of file is_convertible.h.

6.307.2.2 type

```
template<typename T , typename U >
typedef integral_constant<bool,value> Rcpp::traits::is_convertible< T, U >::type
```

Definition at line 41 of file is_convertible.h.

6.307.2.3 value_type

```
template<typename T , typename U >
typedef bool Rcpp::traits::is_convertible< T, U >::value_type
```

Definition at line 39 of file is_convertible.h.

6.307.3 Member Function Documentation

6.307.3.1 MakeT()

```
template<typename T , typename U >
static T Rcpp::traits::is_convertible< T, U >::MakeT ( )  [static], [private]
```

6.307.3.2 Test() [1/2]

```
template<typename T , typename U >
static Big Rcpp::traits::is_convertible< T, U >::Test (
              ...  )  [static], [private]
```

6.307.3.3 Test() [2/2]

```
template<typename T , typename U >
static Small Rcpp::traits::is_convertible< T, U >::Test (
             const U & ) [static], [private]
```

6.307.4 Member Data Documentation

6.307.4.1 value

```
template<typename T , typename U >
const bool Rcpp::traits::is_convertible< T, U >::value = sizeof(Test(MakeT())) == sizeof(Small)
[static]
```

Definition at line 40 of file is_convertible.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/is_convertible.h

6.308 tinyformat::detail::is_convertible< T1, T2 > Struct Template Reference

```
#include <tinyformat.h>
```

Classes

- struct fail
- struct succeed

Static Public Attributes

- static const bool value

Static Private Member Functions

- static fail tryConvert (...)
- static succeed tryConvert (const T2 &)
- static const T1 & makeT1 ()

6.308.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >

Definition at line 176 of file tinyformat.h.

6.308.2 Member Function Documentation

6.308.2.1 makeT1()

```
template<typename T1 , typename T2 >
static const T1& tinyformat::detail::is_convertible< T1, T2 >::makeT1 ( )  [static], [private]
```

6.308.2.2 tryConvert() [1/2]

```
template<typename T1 , typename T2 >
static fail tinyformat::detail::is_convertible< T1, T2 >::tryConvert (
              ...  )  [static], [private]
```

6.308.2.3 tryConvert() [2/2]

```
template<typename T1 , typename T2 >
static succeed tinyformat::detail::is_convertible< T1, T2 >::tryConvert (
             const T2 &  )  [static], [private]
```

6.308.3 Member Data Documentation

6.308.3.1 value

```
template<typename T1 , typename T2 >
const bool tinyformat::detail::is_convertible< T1, T2 >::value  [static]
```

Initial value:

```
=
            sizeof(tryConvert(makeT1())) == sizeof(succeed)
```

Definition at line 197 of file tinyformat.h.

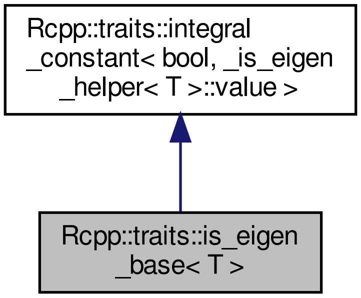
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

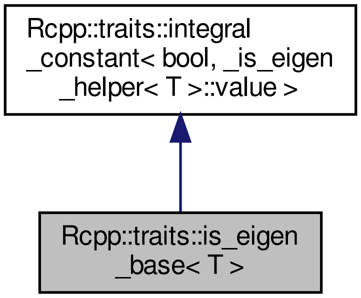
6.309 Rcpp::traits::is_eigen_base< T > Struct Template Reference

```
#include <is_eigen_base.h>
```

Inheritance diagram for Rcpp::traits::is_eigen_base< T >:



Collaboration diagram for Rcpp::traits::is_eigen_base< T >:


Rcpp::traits::integral _constant< bool, _is_eigen _helper< T >::value >
Rcpp::traits::is_eigen _base< T >

Additional Inherited Members

6.309.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_eigen_base< T >

Definition at line 42 of file is_eigen_base.h.

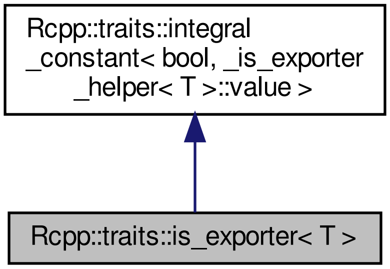
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_eigen_base.h

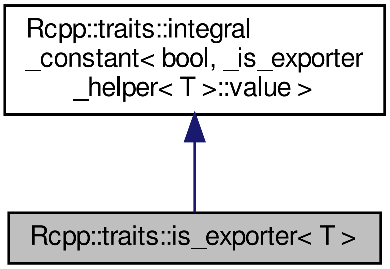
6.310 Rcpp::traits::is_exporter< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is_exporter< T >:



Collaboration diagram for Rcpp::traits::is_exporter< T >:



Additional Inherited Members

6.310.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_exporter< T >

Definition at line 115 of file has_iterator.h.

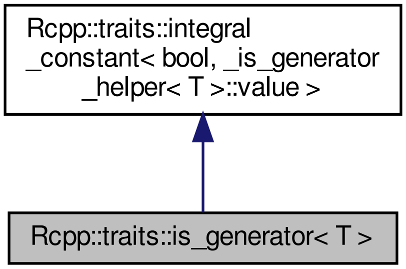
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

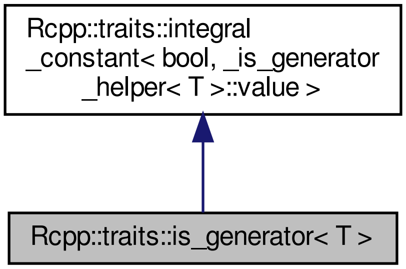
6.311 Rcpp::traits::is_generator< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is_generator< T >:


Rcpp::traits::integral _constant< bool, _is_generator _helper< T >::value >
Rcpp::traits::is_generator< T >

Collaboration diagram for Rcpp::traits::is_generator< T >:


Rcpp::traits::integral _constant< bool, _is_generator _helper< T >::value >
Rcpp::traits::is_generator< T >

Additional Inherited Members

6.311.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_generator< T >

Definition at line 118 of file has_iterator.h.

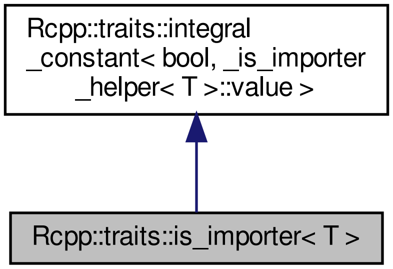
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

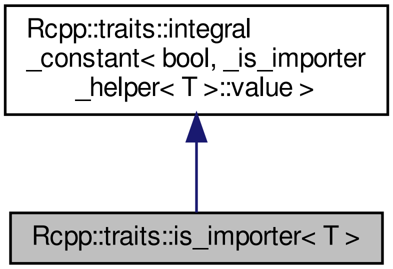
6.312 Rcpp::traits::is_importer< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is_importer< T >:


Rcpp::traits::integral _constant< bool, _is_importer _helper< T >::value >
Rcpp::traits::is_importer< T >

Collaboration diagram for Rcpp::traits::is_importer< T >:


Rcpp::traits::integral _constant< bool, _is_importer _helper< T >::value >
Rcpp::traits::is_importer< T >

Additional Inherited Members

6.312.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_importer< T >

uses SFINAE to identify if a type is importable

The test is based on the presence of a typedef r_import_type in the class

Definition at line 112 of file has_iterator.h.

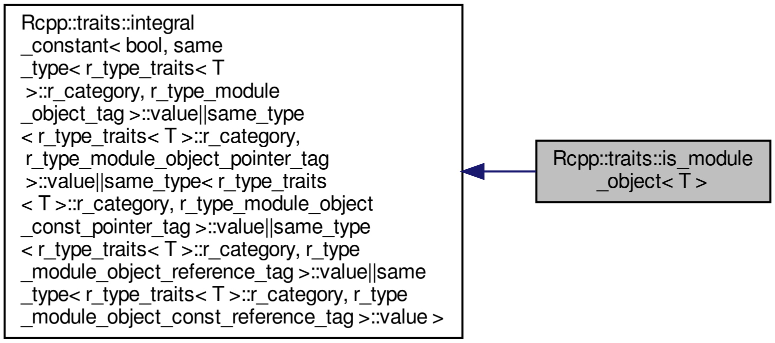
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/has_iterator.h

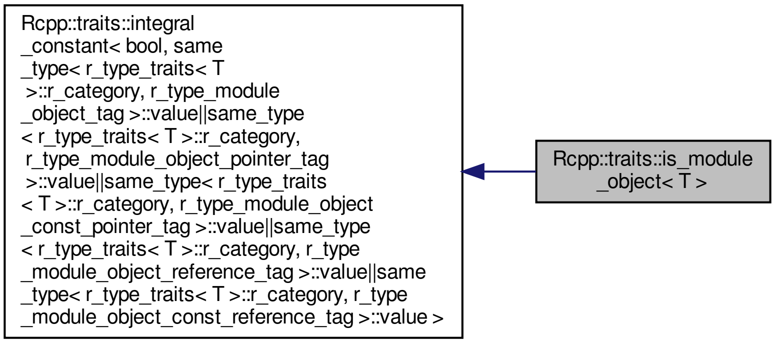
6.313 Rcpp::traits::is_module_object< T > Struct Template Reference

```
#include <is_module_object.h>
```

Inheritance diagram for Rcpp::traits::is_module_object< T >:


Rcpp::traits::integral_constant< bool, same_type< r_type_traits< T >::r_category, r_type_module_object_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_pointer_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_const_pointer_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_reference_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_const_reference_tag >::value >
Rcpp::traits::is_module_object< T >

Collaboration diagram for Rcpp::traits::is_module_object< T >:


Rcpp::traits::integral_constant< bool, same_type< r_type_traits< T >::r_category, r_type_module_object_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_pointer_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_const_pointer_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_reference_tag >::value||same_type< r_type_traits< T >::r_category, r_type_module_object_const_reference_tag >::value >
Rcpp::traits::is_module_object< T >

Additional Inherited Members

6.313.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_module_object< T >

Definition at line 29 of file is_module_object.h.

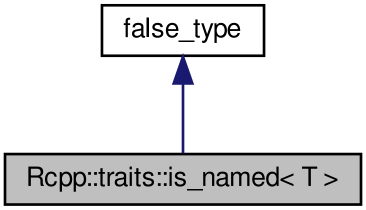
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_module_object.h

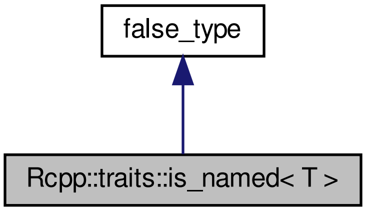
6.314 Rcpp::traits::is_named< T > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::is_named< T >:



Collaboration diagram for Rcpp::traits::is_named< T >:



Additional Inherited Members

6.314.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_named< T >

Definition at line 62 of file named_object.h.

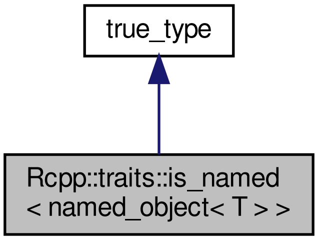
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

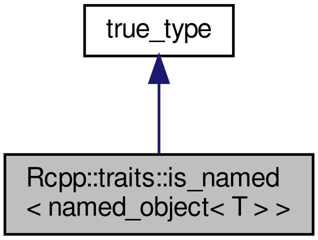
6.315 Rcpp::traits::is_named< named_object< T > > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::is_named< named_object< T > >:



Collaboration diagram for Rcpp::traits::is_named< named_object< T > >:



Additional Inherited Members

6.315.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_named< named_object< T > >

Definition at line 63 of file named_object.h.

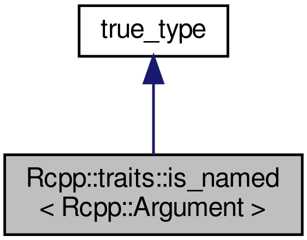
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

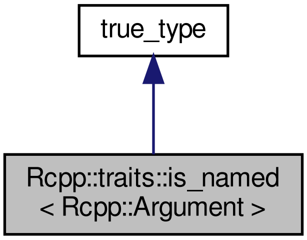
6.316 Rcpp::traits::is_named< Rcpp::Argument > Struct Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::is_named< Rcpp::Argument >:



Collaboration diagram for Rcpp::traits::is_named< Rcpp::Argument >:



Additional Inherited Members

6.316.1 Detailed Description

Definition at line 64 of file named_object.h.

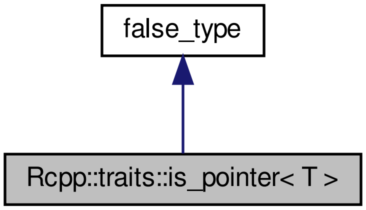
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

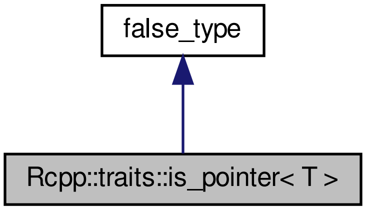
6.317 Rcpp::traits::is_pointer< T > Struct Template Reference

```
#include <is_pointer.h>
```

Inheritance diagram for Rcpp::traits::is_pointer< T >:



Collaboration diagram for Rcpp::traits::is_pointer< T >:



Additional Inherited Members

6.317.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_pointer< T >

Definition at line 29 of file is_pointer.h.

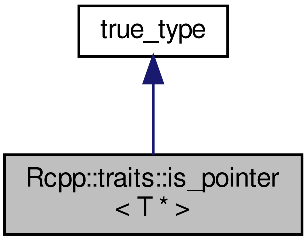
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_pointer.h

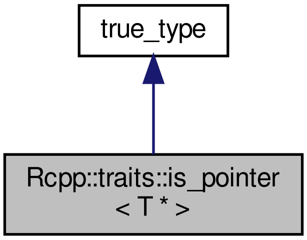
6.318 Rcpp::traits::is_pointer< T ∗ > Struct Template Reference

```
#include <is_pointer.h>
```

Inheritance diagram for Rcpp::traits::is_pointer< T ∗ >:



Collaboration diagram for Rcpp::traits::is_pointer< T ∗ >:



Additional Inherited Members

6.318.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_pointer< T * >

Definition at line 30 of file is_pointer.h.

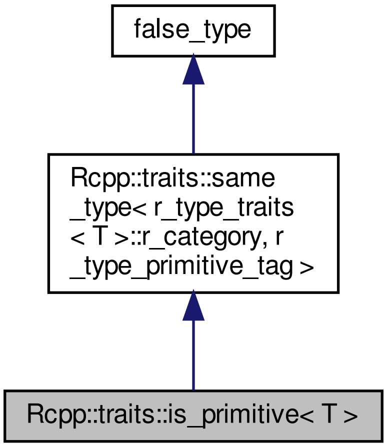
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_pointer.h

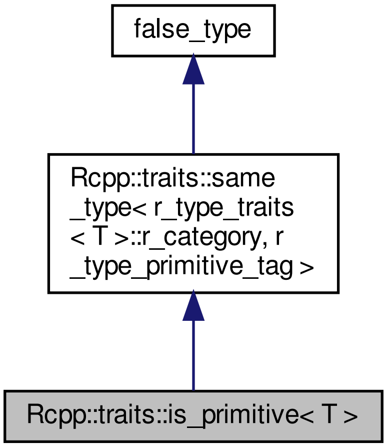
6.319 Rcpp::traits::is_primitive< T > Struct Template Reference

```
#include <is_primitive.h>
```

Inheritance diagram for Rcpp::traits::is_primitive< T >:


false_type
Rcpp::traits::same _type< r_type_traits < T >::r_category, r _type_primitive_tag >
Rcpp::traits::is_primitive< T >

Collaboration diagram for Rcpp::traits::is_primitive< T >:


false_type
Rcpp::traits::same _type< r_type_traits < T >::r_category, r _type_primitive_tag >
Rcpp::traits::is_primitive< T >

Additional Inherited Members

6.319.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_primitive< T >

Definition at line 29 of file is_primitive.h.

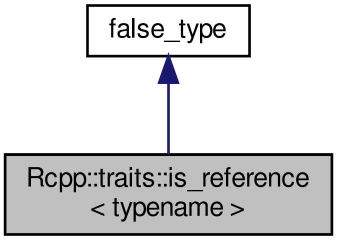
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_primitive.h

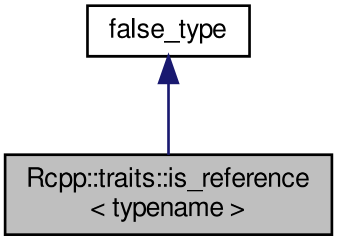
6.320 Rcpp::traits::is_reference< typename > Struct Template Reference

```
#include <is_reference.h>
```

Inheritance diagram for Rcpp::traits::is_reference< typename >:



Collaboration diagram for Rcpp::traits::is_reference< typename >:



Additional Inherited Members

6.320.1 Detailed Description

template<typename>
struct Rcpp::traits::is_reference< typename >

Definition at line 29 of file is_reference.h.

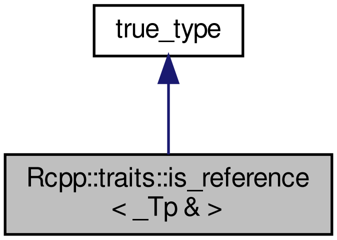
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_reference.h

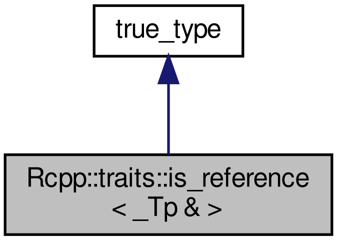
6.321 Rcpp::traits::is_reference< _Tp & > Struct Template Reference

```
#include <is_reference.h>
```

Inheritance diagram for Rcpp::traits::is_reference< _Tp & >:



Collaboration diagram for Rcpp::traits::is_reference< _Tp & >:



Additional Inherited Members

6.321.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::is_reference< _Tp & >

Definition at line 33 of file is_reference.h.

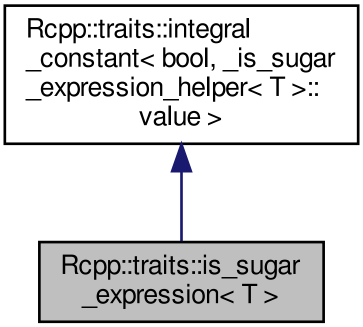
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_reference.h

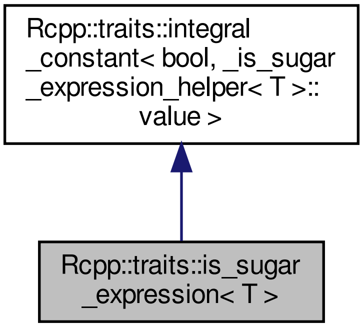
6.322 Rcpp::traits::is_sugar_expression< T > Struct Template Reference

```
#include <is_sugar_expression.h>
```

Inheritance diagram for Rcpp::traits::is_sugar_expression< T >:


Rcpp::traits::integral _constant< bool, _is_sugar _expression_helper< T >:: value >
Rcpp::traits::is_sugar _expression< T >

Collaboration diagram for Rcpp::traits::is_sugar_expression< T >:


Rcpp::traits::integral _constant< bool, _is_sugar _expression_helper< T >:: value >
Rcpp::traits::is_sugar _expression< T >

Additional Inherited Members

6.322.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_sugar_expression< T >

Definition at line 42 of file is_sugar_expression.h.

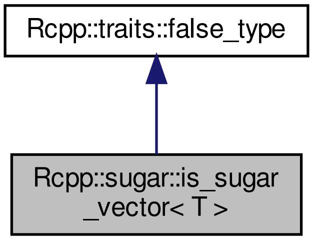
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_sugar_expression.h

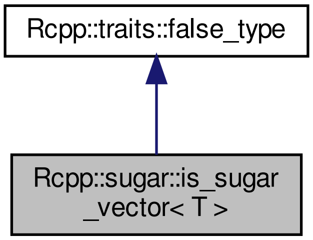
6.323 Rcpp::sugar::is_sugar_vector< T > Struct Template Reference

```
#include <iterator.h>
```

Inheritance diagram for Rcpp::sugar::is_sugar_vector< T >:



Collaboration diagram for Rcpp::sugar::is_sugar_vector< T >:



Additional Inherited Members

6.323.1 Detailed Description

template<typename T>
struct Rcpp::sugar::is_sugar_vector< T >

Definition at line 121 of file iterator.h.

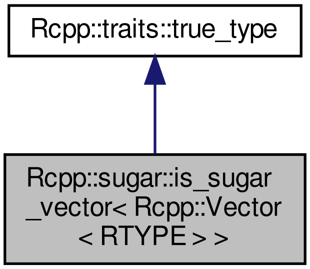
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

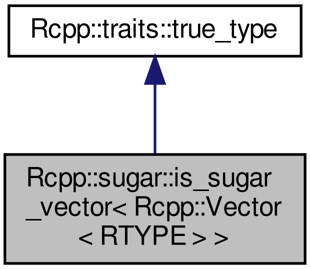
6.324 Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > > Struct Template Reference

```
#include <iterator.h>
```

Inheritance diagram for Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >:



Collaboration diagram for Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >:



Additional Inherited Members

6.324.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >

Definition at line 122 of file iterator.h.

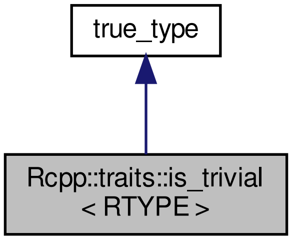
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

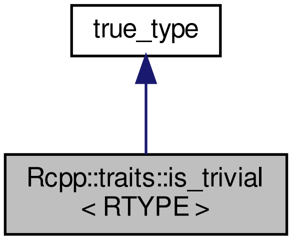
6.325 Rcpp::traits::is_trivial< RTYPE > Struct Template Reference

```
#include <is_trivial.h>
```

Inheritance diagram for Rcpp::traits::is_trivial< RTYPE >:



Collaboration diagram for Rcpp::traits::is_trivial< RTYPE >:



Additional Inherited Members

6.325.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::is_trivial< RTYPE >

Definition at line 28 of file is_trivial.h.

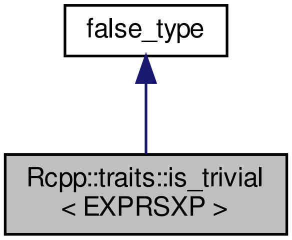
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_trivial.h

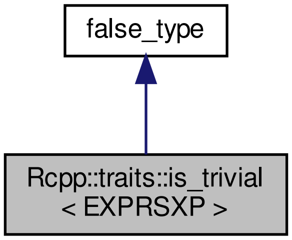
6.326 Rcpp::traits::is_trivial< EXPRSXP > Struct Reference

```
#include <is_trivial.h>
```

Inheritance diagram for Rcpp::traits::is_trivial< EXPRSXP >:



Collaboration diagram for Rcpp::traits::is_trivial< EXPRSXP >:



Additional Inherited Members

6.326.1 Detailed Description

Definition at line 30 of file is_trivial.h.

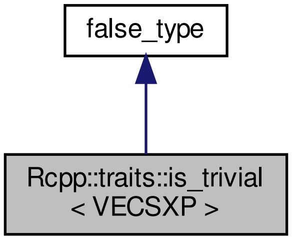
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_trivial.h

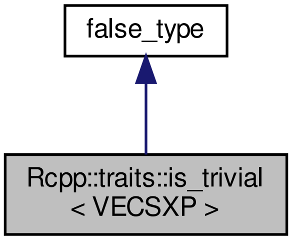
6.327 Rcpp::traits::is_trivial< VECSXP > Struct Reference

```
#include <is_trivial.h>
```

Inheritance diagram for Rcpp::traits::is_trivial< VECSXP >:



Collaboration diagram for Rcpp::traits::is_trivial< VECSXP >:



Additional Inherited Members

6.327.1 Detailed Description

Definition at line 29 of file is_trivial.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_trivial.h

6.328 tinyformat::detail::is_wchar< T > Struct Template Reference

```
#include <tinyformat.h>
```

Public Types

- typedef int tinyformat_wchar_is_not_supported

6.328.1 Detailed Description

template<typename T>
struct tinyformat::detail::is_wchar< T >

Definition at line 206 of file tinyformat.h.

6.328.2 Member Typedef Documentation

6.328.2.1 tinyformat_wchar_is_not_supported

```
template<typename T >
typedef int tinyformat::detail::is_wchar< T >::tinyformat_wchar_is_not_supported
```

Definition at line 206 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.329 tinyformat::detail::is_wchar< const wchar_t * > Struct Reference

```
#include <tinyformat.h>
```

6.329.1 Detailed Description

Definition at line 208 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.330 tinyformat::detail::is_wchar< const wchar_t[n]> Struct Template Reference

```
#include <tinyformat.h>
```

6.330.1 Detailed Description

template<int n>
struct tinyformat::detail::is_wchar< const wchar_t[n]>

Definition at line 209 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.331 tinyformat::detail::is_wchar< wchar_t * > Struct Reference

```
#include <tinyformat.h>
```

6.331.1 Detailed Description

Definition at line 207 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.332 tinyformat::detail::is_wchar< wchar_t[n]> Struct Template Reference

```
#include <tinyformat.h>
```

6.332.1 Detailed Description

template<int n>
struct tinyformat::detail::is_wchar< wchar_t[n]>

Definition at line 210 of file tinyformat.h.

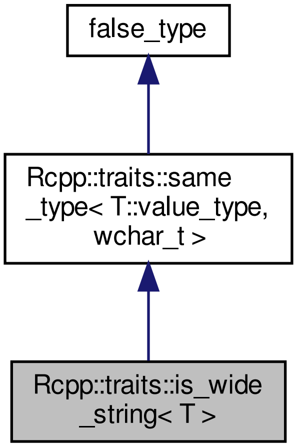
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

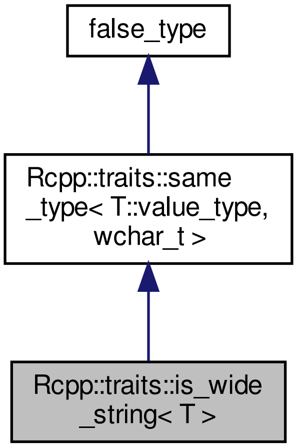
6.333 Rcpp::traits::is_wide_string< T > Struct Template Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is_wide_string< T >:


false_type
Rcpp::traits::same_type< T::value_type, wchar_t >
Rcpp::traits::is_wide_string< T >

Collaboration diagram for Rcpp::traits::is_wide_string< T >:


false_type
Rcpp::traits::same_type< T::value_type, wchar_t >
Rcpp::traits::is_wide_string< T >

Additional Inherited Members

6.333.1 Detailed Description

template<typename T>
struct Rcpp::traits::is_wide_string< T >

Definition at line 30 of file is_wide_string.h.

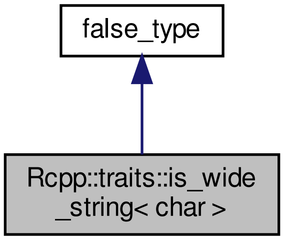
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

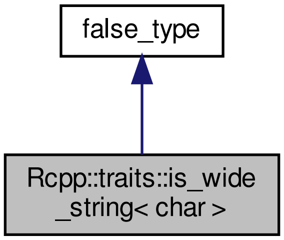
6.334 Rcpp::traits::is_wide_string< char > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is_wide_string< char >:



Collaboration diagram for Rcpp::traits::is_wide_string< char >:



Additional Inherited Members

6.334.1 Detailed Description

Definition at line 36 of file is_wide_string.h.

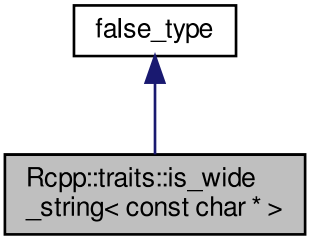
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

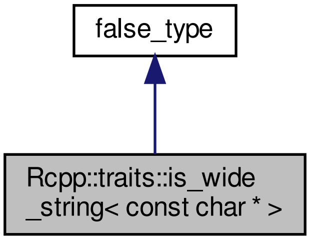
6.335 Rcpp::traits::is_wide_string< const char * > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is_wide_string< const char * >:



Collaboration diagram for Rcpp::traits::is_wide_string< const char * >:



Additional Inherited Members

6.335.1 Detailed Description

Definition at line 33 of file is_wide_string.h.

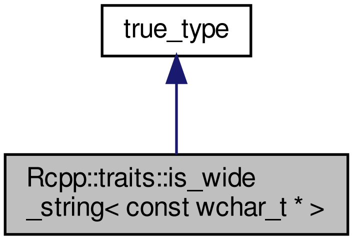
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

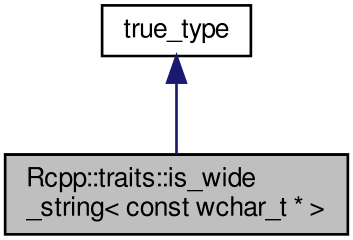
6.336 Rcpp::traits::is_wide_string< const wchar_t * > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is_wide_string< const wchar_t * >:



Collaboration diagram for Rcpp::traits::is_wide_string< const wchar_t * >:



Additional Inherited Members

6.336.1 Detailed Description

Definition at line 32 of file is_wide_string.h.

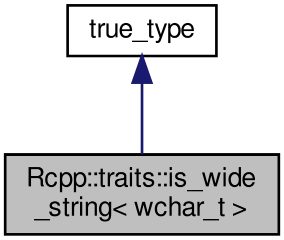
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

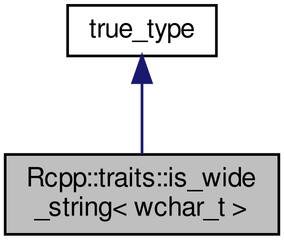
6.337 Rcpp::traits::is_wide_string< wchar_t > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is_wide_string< wchar_t >:



Collaboration diagram for Rcpp::traits::is_wide_string< wchar_t >:



Additional Inherited Members

6.337.1 Detailed Description

Definition at line 35 of file is_wide_string.h.

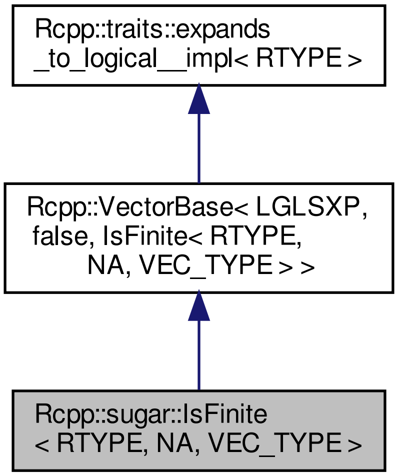
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is_wide_string.h

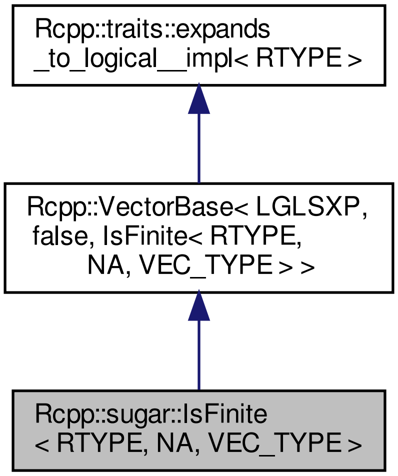
6.338 Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE > Class Template Reference

```
#include <is_finite.h>
```

Inheritance diagram for Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >:



Collaboration diagram for Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, IsFinite< RTYPE, NA, VEC_TYPE > >
Rcpp::sugar::IsFinite < RTYPE, NA, VEC_TYPE >

Public Member Functions

- IsFinite (const VEC_TYPE &obj_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & obj

Additional Inherited Members

6.338.1 Detailed Description

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >

Definition at line 29 of file is_finite.h.

6.338.2 Constructor & Destructor Documentation

6.338.2.1 IsFinite()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::IsFinite (
             const VEC_TYPE & obj_ )  [inline]
```

Definition at line 32 of file is_finite.h.

6.338.3 Member Function Documentation

6.338.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 34 of file is_finite.h.

References Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj.

6.338.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::size ( ) const  [inline]
```

Definition at line 38 of file is_finite.h.

References Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj.

6.338.4 Member Data Documentation

6.338.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj  [private]
```

Definition at line 41 of file is_finite.h.

Referenced by Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::operator[](), and Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::size().

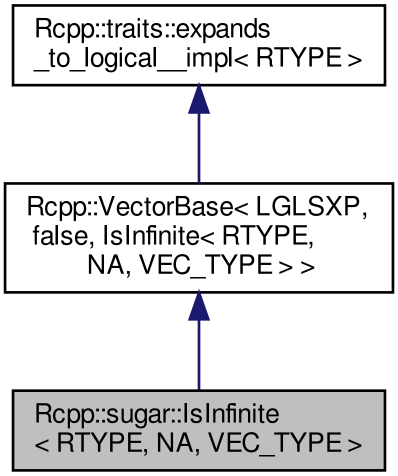
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_finite.h

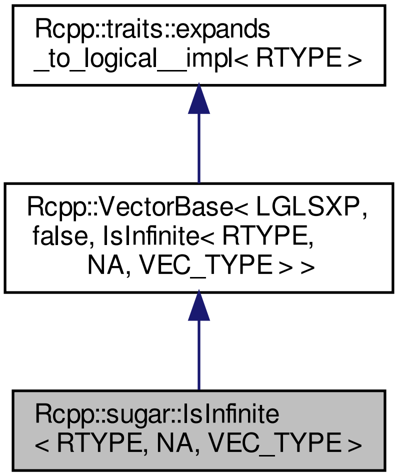
6.339 Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE > Class Template Reference

```
#include <is_infinite.h>
```

Inheritance diagram for Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, IsInfinite< RTYPE, NA, VEC_TYPE > >
Rcpp::sugar::IsInfinite < RTYPE, NA, VEC_TYPE >

Collaboration diagram for Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >:



Public Member Functions

- IsInfinite (const VEC_TYPE &obj_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & obj

Additional Inherited Members

6.339.1 Detailed Description

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >

Definition at line 29 of file is_infinite.h.

6.339.2 Constructor & Destructor Documentation

6.339.2.1 IsInfinite()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::IsInfinite (
             const VEC_TYPE & obj_ )  [inline]
```

Definition at line 32 of file is_infinite.h.

6.339.3 Member Function Documentation

6.339.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 34 of file is_infinite.h.

References Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj.

6.339.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::size ( ) const  [inline]
```

Definition at line 38 of file is_infinite.h.

References Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj.

6.339.4 Member Data Documentation

6.339.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj  [private]
```

Definition at line 41 of file is_infinite.h.

Referenced by Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::operator[ ](), and Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::size().

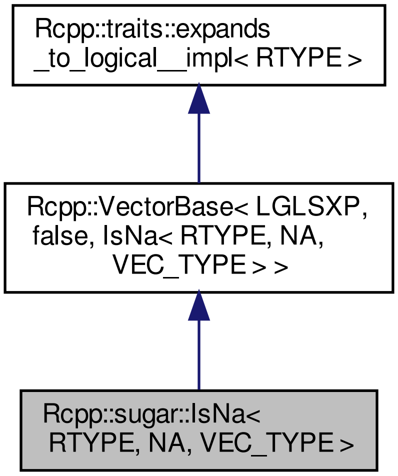
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_infinite.h

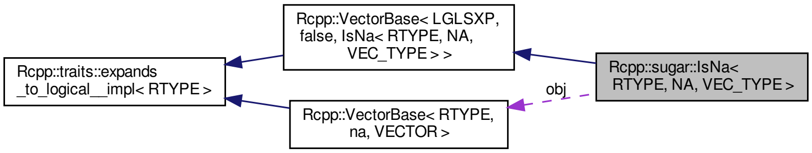
6.340 Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > >
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >

Collaboration diagram for Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
obj
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, NA, VEC_TYPE > BASE

Public Member Functions

- IsNa (const BASE &obj_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const BASE & obj

6.340.1 Detailed Description

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >

Definition at line 29 of file is_na.h.

6.340.2 Member Typedef Documentation

6.340.2.1 BASE

```
template<int RTYPE, bool NA, typename VEC_TYPE >
typedef Rcpp::VectorBase<RTYPE,NA,VEC_TYPE> Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::BASE
```

Definition at line 32 of file is_na.h.

6.340.2.2 STORAGE

```
template<int RTYPE, bool NA, typename VEC_TYPE >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::STORAGE
```

Definition at line 31 of file is_na.h.

6.340.3 Constructor & Destructor Documentation

6.340.3.1 IsNa()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::IsNa (
            const BASE & obj_ ) [inline]
```

Definition at line 34 of file is_na.h.

6.340.4 Member Function Documentation

6.340.4.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 36 of file is_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::obj.

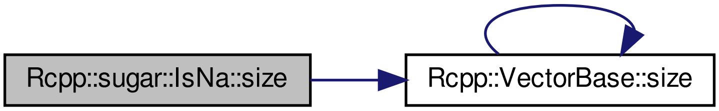
6.340.4.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 40 of file is_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.340.5 Member Data Documentation

6.340.5.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const BASE& Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::obj  [private]
```

Definition at line 43 of file is_na.h.

Referenced by Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::operator[ ](), Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size(), and Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size().

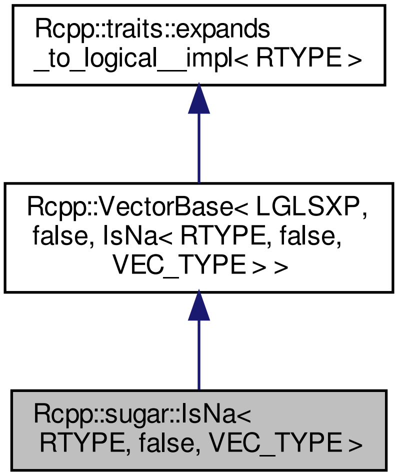
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

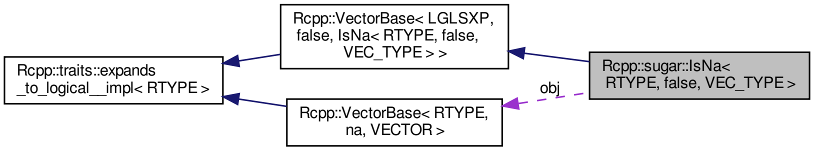
6.341 Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP,
false, IsNa< RTYPE, false,
VEC_TYPE > >
Rcpp::sugar::IsNa<
RTYPE, false, VEC_TYPE >

Collaboration diagram for Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >:


Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, false, VEC_TYPE > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >
obj
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, VEC_TYPE > BASE

Public Member Functions

- IsNa (const BASE &obj_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const BASE & obj

6.341.1 Detailed Description

template<int RTYPE, typename VEC_TYPE>
class Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >

Definition at line 51 of file is_na.h.

6.341.2 Member Typedef Documentation

6.341.2.1 BASE

```
template<int RTYPE, typename VEC_TYPE >
typedef Rcpp::VectorBase<RTYPE,false,VEC_TYPE> Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::BASE
```

Definition at line 54 of file is_na.h.

6.341.2.2 STORAGE

```
template<int RTYPE, typename VEC_TYPE >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::STORAGE
```

Definition at line 53 of file is_na.h.

6.341.3 Constructor & Destructor Documentation

6.341.3.1 IsNa()

```
template<int RTYPE, typename VEC_TYPE >
Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::IsNa (
             const BASE & obj_ )  [inline]
```

Definition at line 56 of file is_na.h.

6.341.4 Member Function Documentation

6.341.4.1 operator[]()

```
template<int RTYPE, typename VEC_TYPE >
int Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 58 of file is_na.h.

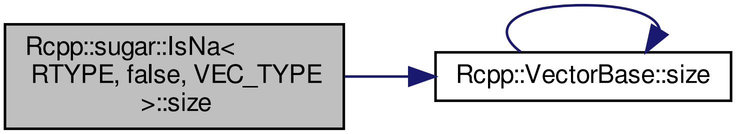
6.341.4.2 size()

```
template<int RTYPE, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size ( ) const  [inline]
```

Definition at line 62 of file is_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >↩ ::size().

Here is the call graph for this function:



6.341.5 Member Data Documentation

6.341.5.1 obj

```
template<int RTYPE, typename VEC_TYPE >
const BASE& Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::obj  [private]
```

Definition at line 65 of file is_na.h.

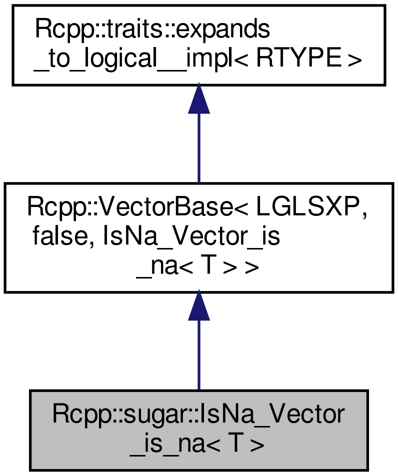
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

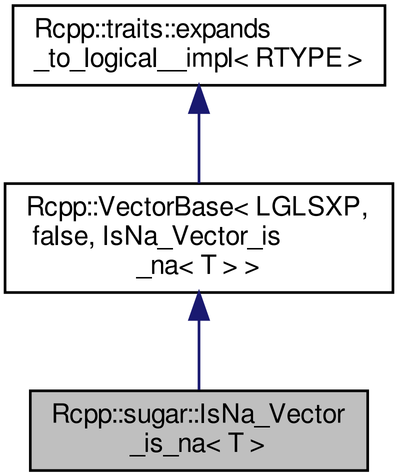
6.342 Rcpp::sugar::IsNa_Vector_is_na< T > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for Rcpp::sugar::IsNa_Vector_is_na< T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, IsNa_Vector_is_na< T > >
Rcpp::sugar::IsNa_Vector_is_na< T >

Collaboration diagram for Rcpp::sugar::IsNa_Vector_is_na< T >:



Public Member Functions

- IsNa_Vector_is_na (const T &x)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T & ref

Additional Inherited Members

6.342.1 Detailed Description

template<typename T>
class Rcpp::sugar::IsNa_Vector_is_na< T >

Definition at line 70 of file is_na.h.

6.342.2 Constructor & Destructor Documentation

6.342.2.1 IsNa_Vector_is_na()

```
template<typename T >
Rcpp::sugar::IsNa_Vector_is_na< T >::IsNa_Vector_is_na (
             const T & x )  [inline]
```

Definition at line 72 of file is_na.h.

6.342.3 Member Function Documentation

6.342.3.1 operator[]()

```
template<typename T >
int Rcpp::sugar::IsNa_Vector_is_na< T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 74 of file is_na.h.

References Rcpp::sugar::IsNa_Vector_is_na< T >::ref.

6.342.3.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::IsNa_Vector_is_na< T >::size ( ) const  [inline]
```

Definition at line 82 of file is_na.h.

References Rcpp::sugar::IsNa_Vector_is_na< T >::ref.

6.342.4 Member Data Documentation

6.342.4.1 ref

```
template<typename T >
const T& Rcpp::sugar::IsNa_Vector_is_na< T >::ref  [private]
```

Definition at line 85 of file is_na.h.

Referenced by Rcpp::sugar::IsNa_Vector_is_na< T >::operator[ ](), and Rcpp::sugar::IsNa_Vector_is_na< T >::size().

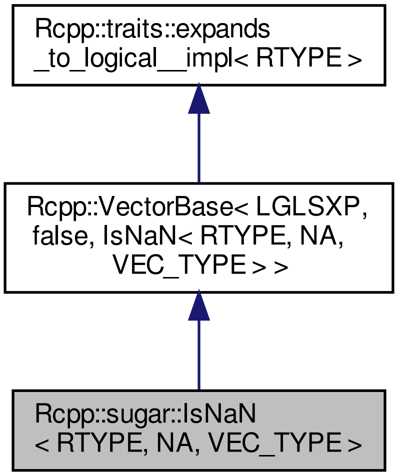
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_na.h

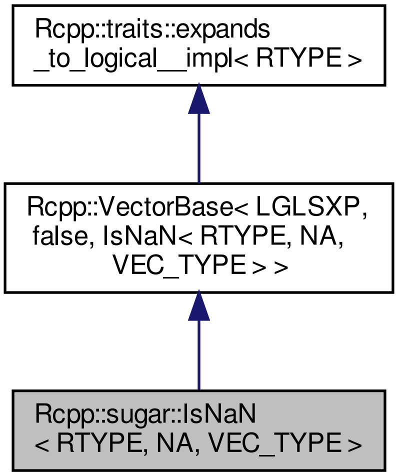
6.343 Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE > Class Template Reference

```
#include <is_nan.h>
```

Inheritance diagram for Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >:



Collaboration diagram for Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >:



Public Member Functions

- IsNaN (const VEC_TYPE &obj_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & obj

Additional Inherited Members

6.343.1 Detailed Description

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >

Definition at line 29 of file is_nan.h.

6.343.2 Constructor & Destructor Documentation

6.343.2.1 IsNaN()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::IsNaN (
             const VEC_TYPE & obj_ )  [inline]
```

Definition at line 32 of file is_nan.h.

6.343.3 Member Function Documentation

6.343.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 34 of file is_nan.h.

References Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj.

6.343.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::size ( ) const  [inline]
```

Definition at line 38 of file is_nan.h.

References Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj.

6.343.4 Member Data Documentation

6.343.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj  [private]
```

Definition at line 41 of file is_nan.h.

Referenced by Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::operator[ ](), and Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::size().

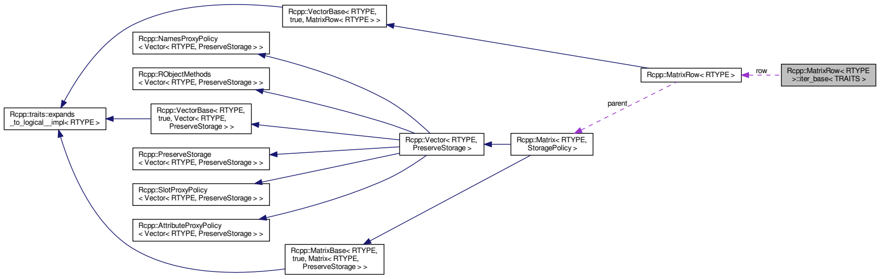
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/is_nan.h

6.344 Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS > Class Template Reference

```
#include <MatrixRow.h>
```

Collaboration diagram for Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >:


Rcpp::VectorBase< RTYPE, true, MatrixRow< RTYPE > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
parent
Rcpp::MatrixRow< RTYPE >
row
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >

Public Types

- typedef TRAITS::vector_iterator vector_iterator
- typedef TRAITS::difference_type difference_type
- typedef TRAITS::value_type value_type
- typedef TRAITS::reference reference
- typedef TRAITS::pointer pointer
- typedef std::random_access_iterator_tag iterator_category

Public Member Functions

- iter_base (const iter_base &other)
- iter_base (MatrixRow &row_, int index_)
- iter_base & operator++ ()
- iter_base operator++ (int)
- iter_base & operator-- ()
- iter_base operator-- (int)
- iter_base operator+ (difference_type n) const
- iter_base operator- (difference_type n) const
- difference_type operator- (const iter_base &other) const
- iter_base & operator+= (difference_type n)
- iter_base & operator-= (difference_type n)
- reference operator∗ ()
- pointer operator-> ()
- bool operator== (const iter_base &other)
- bool operator!= (const iter_base &other)
- bool operator< (const iter_base &other)
- bool operator> (const iter_base &other)
- bool operator<= (const iter_base &other)
- bool operator>= (const iter_base &other)
- reference operator[ ] (int i) const
- difference_type operator- (const iter_base &other)

Private Attributes

- MatrixRow & row
- int index

6.344.1 Detailed Description

template<int RTYPE>
template<typename TRAITS>
class Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >

Definition at line 62 of file MatrixRow.h.

6.344.2 Member Typedef Documentation

6.344.2.1 difference_type

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::difference_type
```

Definition at line 66 of file MatrixRow.h.

6.344.2.2 iterator_category

```
template<int RTYPE>
template<typename TRAITS >
typedef std::random_access_iterator_tag Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iterator_category
```

Definition at line 71 of file MatrixRow.h.

6.344.2.3 pointer

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::pointer Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::pointer
```

Definition at line 69 of file MatrixRow.h.

6.344.2.4 reference

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::reference
```

Definition at line 68 of file MatrixRow.h.

6.344.2.5 value_type

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::value_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::value_type
```

Definition at line 67 of file MatrixRow.h.

6.344.2.6 vector_iterator

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::vector_iterator Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::vector_iterator
```

Definition at line 64 of file MatrixRow.h.

6.344.3 Constructor & Destructor Documentation

6.344.3.1 iter_base() [1/2]

```
template<int RTYPE>
template<typename TRAITS >
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 73 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+(), and Rcpp::MatrixRow< RTYPE >↩
::iter_base< TRAITS >::operator-().

6.344.3.2 iter_base() [2/2]

```
template<int RTYPE>
template<typename TRAITS >
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base (
             MatrixRow & row_,
             int index_ )  [inline]
```

Definition at line 74 of file MatrixRow.h.

6.344.4 Member Function Documentation

6.344.4.1 operator"!=()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator!= (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 111 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.2 operator∗()

```
template<int RTYPE>
template<typename TRAITS >
reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator* ( )  [inline]
```

Definition at line 103 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter_↩ base< TRAITS >::row.

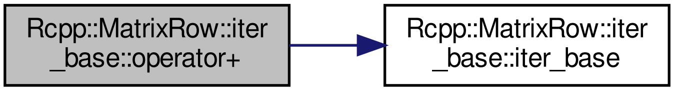
6.344.4.3 operator+()

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+ (
            difference_type n ) const  [inline]
```

Definition at line 96 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index, Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base(), and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row.

Here is the call graph for this function:



6.344.4.4 operator++() [1/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator++ ( )  [inline]
```

Definition at line 76 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.5 operator++() [2/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator++ (
            int  )  [inline]
```

Definition at line 80 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.6 operator+=()

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+= (
             difference_type n )  [inline]
```

Definition at line 100 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.7 operator-() [1/3]

```
template<int RTYPE>
template<typename TRAITS >
difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 121 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.8 operator-() [2/3]

```
template<int RTYPE>
template<typename TRAITS >
difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
             const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 98 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

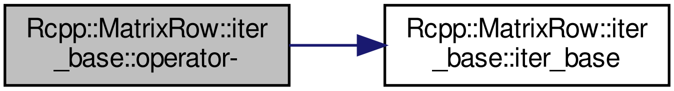
6.344.4.9 operator-() [3/3]

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
             difference_type n ) const  [inline]
```

Definition at line 97 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index, Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base(), and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row.

Here is the call graph for this function:


Rcpp::MatrixRow::iter_base::operator-
Rcpp::MatrixRow::iter_base::iter_base

6.344.4.10 operator--() [1/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-- ( )  [inline]
```

Definition at line 86 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.11 operator--() [2/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-- (
             int  )  [inline]
```

Definition at line 90 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.12 operator-=()

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-= (
             difference_type n )  [inline]
```

Definition at line 101 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.13 operator->()

```
template<int RTYPE>
template<typename TRAITS >
pointer Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-> ( )  [inline]
```

Definition at line 106 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row.

6.344.4.14 operator<()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator< (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 112 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.15 operator<=()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator<= (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 114 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.16 operator==()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator== (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 110 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.17 operator>()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator> (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 113 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.18 operator>=()

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator>= (
             const iter_base< TRAITS > & other )  [inline]
```

Definition at line 115 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index.

6.344.4.19 operator[]()

```
template<int RTYPE>
template<typename TRAITS >
reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator[] (
             int i ) const  [inline]
```

Definition at line 117 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row.

6.344.5 Member Data Documentation

6.344.5.1 index

```
template<int RTYPE>
template<typename TRAITS >
int Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index  [private]
```

Definition at line 127 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator!=(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator∗(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator++(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+=(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator--(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-=(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator->(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator<(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator<=(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator==(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator>(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator>=(), and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator[ ]().

6.344.5.2 row

```
template<int RTYPE>
template<typename TRAITS >
MatrixRow& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row  [private]
```

Definition at line 126 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator∗(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-(), Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator->(), and Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.345 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS > Class Template Reference

```
#include <VectorBase.h>
```

Public Types

- typedef TRAITS::reference reference
- typedef TRAITS::pointer pointer
- typedef TRAITS::difference_type difference_type
- typedef TRAITS::value_type value_type
- typedef TRAITS::iterator_category iterator_category

Public Member Functions

- iter_base (const VectorBase &object_, R_xlen_t index_)
- iter_base & operator++ ()
- iter_base operator++ (int)
- iter_base & operator-- ()
- iter_base operator-- (int)
- iter_base operator+ (difference_type n) const
- iter_base operator- (difference_type n) const
- iter_base & operator+= (difference_type n)
- iter_base & operator-= (difference_type n)
- reference operator[ ] (R_xlen_t i)
- reference operator∗ ()
- pointer operator-> ()
- bool operator== (const iter_base &y) const
- bool operator!= (const iter_base &y) const
- bool operator< (const iter_base &other) const
- bool operator> (const iter_base &other) const
- bool operator<= (const iter_base &other) const
- bool operator>= (const iter_base &other) const
- difference_type operator- (const iter_base &other) const

Private Attributes

- const VECTOR & object
- R_xlen_t index

6.345.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
template<typename TRAITS>
class Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >

Definition at line 75 of file VectorBase.h.

6.345.2 Member Typedef Documentation

6.345.2.1 difference_type

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::difference_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >↩
::difference_type
```

Definition at line 79 of file VectorBase.h.

6.345.2.2 iterator_category

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::iterator_category Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >↩
::iterator_category
```

Definition at line 81 of file VectorBase.h.

6.345.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::pointer Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::pointer
```

Definition at line 78 of file VectorBase.h.

6.345.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::reference
```

Definition at line 77 of file VectorBase.h.

6.345.2.5 value_type

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::value_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::value_type
```

Definition at line 80 of file VectorBase.h.

6.345.3 Constructor & Destructor Documentation

6.345.3.1 iter_base()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::iter_base (
             const VectorBase & object_,
             R_xlen_t index_ )  [inline]
```

Definition at line 83 of file VectorBase.h.

6.345.4 Member Function Documentation

6.345.4.1 operator"!=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator!= (
             const iter_base< TRAITS > & y ) const  [inline]
```

Definition at line 135 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.2 operator∗()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator∗ ( )  [inline]
```

Definition at line 125 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.3 operator+()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+ (
             difference_type n ) const  [inline]
```

Definition at line 105 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.4 operator++() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator++ ( )  [inline]
```

Definition at line 85 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.5 operator++() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator++ (
             int  )  [inline]
```

Definition at line 89 of file VectorBase.h.

6.345.4.6 operator+=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+= (
             difference_type n )  [inline]
```

Definition at line 112 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.7 operator-() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
difference_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator- (
             const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 151 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.8 operator-() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator- (
             difference_type n ) const  [inline]
```

Definition at line 108 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.9 operator--() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-- ( )  [inline]
```

Definition at line 95 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.10 operator--() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-- (
             int  )  [inline]
```

Definition at line 99 of file VectorBase.h.

6.345.4.11 operator-=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-= (
             difference_type n )  [inline]
```

Definition at line 116 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.12 operator->()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
pointer Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-> ( )  [inline]
```

Definition at line 128 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.13 operator<()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator< (
             const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 138 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.14 operator<=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator<= (
             const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 144 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.15 operator==()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator== (
            const iter_base< TRAITS > & y ) const  [inline]
```

Definition at line 132 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.16 operator>()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator> (
            const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 141 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.17 operator>=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator>= (
            const iter_base< TRAITS > & other ) const  [inline]
```

Definition at line 147 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.4.18 operator[]()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator[] (
            R_xlen_t i )  [inline]
```

Definition at line 121 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index.

6.345.5 Member Data Documentation

6.345.5.1 index

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index  [private]
```

Definition at line 158 of file VectorBase.h.

Referenced by Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator!=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator∗(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator++(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator--(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator->(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator<(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator<=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator==(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator>(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator>=(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator[ ]().

6.345.5.2 object

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
const VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::object  [private]
```

Definition at line 157 of file VectorBase.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

6.346 Rcpp::MatrixRow< RTYPE >::iter_traits Struct Reference

```
#include <MatrixRow.h>
```

Public Types

- typedef traits::r_vector_iterator< RTYPE >::type vector_iterator
- typedef int difference_type
- typedef traits::r_vector_proxy< RTYPE >::type value_type
- typedef value_type reference
- typedef std::iterator_traits< vector_iterator >::pointer pointer

6.346.1 Detailed Description

template<int RTYPE>
struct Rcpp::MatrixRow< RTYPE >::iter_traits

Definition at line 41 of file MatrixRow.h.

6.346.2 Member Typedef Documentation

6.346.2.1 difference_type

```
template<int RTYPE>
typedef int Rcpp::MatrixRow< RTYPE >::iter_traits::difference_type
```

Definition at line 45 of file MatrixRow.h.

6.346.2.2 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector_iterator>::pointer Rcpp::MatrixRow< RTYPE >::iter_traits::pointer
```

Definition at line 48 of file MatrixRow.h.

6.346.2.3 reference

```
template<int RTYPE>
typedef value_type Rcpp::MatrixRow< RTYPE >::iter_traits::reference
```

Definition at line 47 of file MatrixRow.h.

6.346.2.4 value_type

```
template<int RTYPE>
typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iter_traits::value_type
```

Definition at line 46 of file MatrixRow.h.

6.346.2.5 vector_iterator

```
template<int RTYPE>
typedef traits::r_vector_iterator<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iter_traits::vector_iterator
```

Definition at line 43 of file MatrixRow.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.347 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits Struct Reference

```
#include <VectorBase.h>
```

Public Types

- typedef stored_type & reference
- typedef stored_type ∗ pointer
- typedef R_xlen_t difference_type
- typedef stored_type value_type
- typedef std::random_access_iterator_tag iterator_category

6.347.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits

Definition at line 56 of file VectorBase.h.

6.347.2 Member Typedef Documentation

6.347.2.1 difference_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::difference_type
```

Definition at line 60 of file VectorBase.h.

6.347.2.2 iterator_category

```
template<int RTYPE, bool na, typename VECTOR >
typedef std::random_access_iterator_tag Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::iterator_category
```

Definition at line 62 of file VectorBase.h.

6.347.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type* Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::pointer
```

Definition at line 59 of file VectorBase.h.

6.347.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::reference
```

Definition at line 58 of file VectorBase.h.

6.347.2.5 value_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::value_type
```

Definition at line 61 of file VectorBase.h.

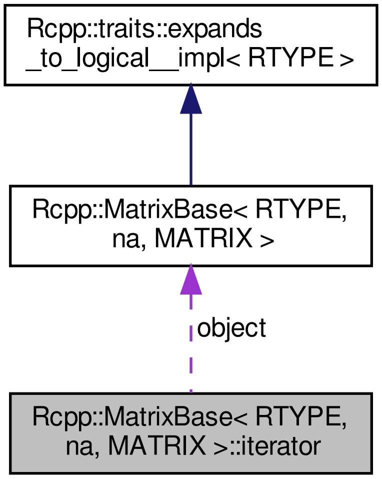
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

6.348 Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator Class Reference

```
#include <MatrixBase.h>
```

Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, na, MATRIX >
object
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator

Public Types

- typedef stored_type reference
- typedef stored_type ∗ pointer
- typedef R_xlen_t difference_type
- typedef stored_type value_type
- typedef std::random_access_iterator_tag iterator_category

Public Member Functions

- iterator (const MatrixBase &object_, R_xlen_t index_)
- iterator (const iterator &other)
- iterator & operator++ ()
- iterator operator++ (int)
- iterator & operator-- ()
- iterator operator-- (int)
- iterator operator+ (difference_type n) const
- iterator operator- (difference_type n) const
- iterator & operator+= (difference_type n)
- iterator & operator-= (difference_type n)

- reference operator* ()
- pointer operator-> ()
- bool operator== (const iterator &y) const
- bool operator!= (const iterator &y) const
- bool operator< (const iterator &other) const
- bool operator> (const iterator &other) const
- bool operator<= (const iterator &other) const
- bool operator>= (const iterator &other) const
- difference_type operator- (const iterator &other) const

Private Member Functions

- void update_index (int index_)
- R_xlen_t index () const

Private Attributes

- const MatrixBase & object
- int i
- int j
- int nr
- int nc

6.348.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
class Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator

Definition at line 83 of file MatrixBase.h.

6.348.2 Member Typedef Documentation

6.348.2.1 difference_type

```
template<int RTYPE, bool na, typename MATRIX >
typedef R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::difference_type
```

Definition at line 87 of file MatrixBase.h.

6.348.2.2 iterator_category

```
template<int RTYPE, bool na, typename MATRIX >
typedef std::random_access_iterator_tag Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator_category
```

Definition at line 89 of file MatrixBase.h.

6.348.2.3 pointer

```
template<int RTYPE, bool na, typename MATRIX >
typedef stored_type* Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::pointer
```

Definition at line 86 of file MatrixBase.h.

6.348.2.4 reference

```
template<int RTYPE, bool na, typename MATRIX >
typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::reference
```

Definition at line 85 of file MatrixBase.h.

6.348.2.5 value_type

```
template<int RTYPE, bool na, typename MATRIX >
typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::value_type
```

Definition at line 88 of file MatrixBase.h.

6.348.3 Constructor & Destructor Documentation

6.348.3.1 iterator() [1/2]

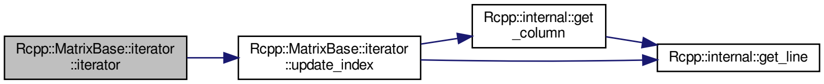
```
template<int RTYPE, bool na, typename MATRIX >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator (
             const MatrixBase & object_,
             R_xlen_t index_ )  [inline]
```

Definition at line 91 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::iterator
Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

6.348.3.2 iterator() [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator (
             const iterator & other )  [inline]
```

Definition at line 97 of file MatrixBase.h.

6.348.4 Member Function Documentation

6.348.4.1 index()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index ( ) const  [inline], [private]
```

Definition at line 185 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>=().

6.348.4.2 operator"!=()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!= (
             const iterator & y ) const  [inline]
```

Definition at line 154 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j.

6.348.4.3 operator∗()

```
template<int RTYPE, bool na, typename MATRIX >
reference Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator* ( )  [inline]
```

Definition at line 144 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object.

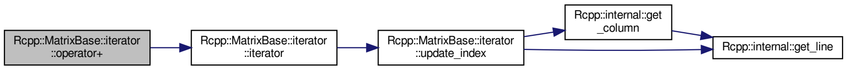
6.348.4.4 operator+()

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+ (
             difference_type n ) const  [inline]
```

Definition at line 128 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator+
Rcpp::MatrixBase::iterator ::iterator
Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

6.348.4.5 operator++() [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ ( )  [inline]
```

Definition at line 100 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

6.348.4.6 operator++() [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ (
             int  )  [inline]
```

Definition at line 108 of file MatrixBase.h.

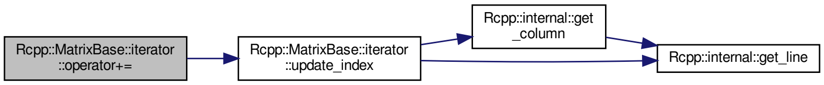
6.348.4.7 operator+=()

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+= (
             difference_type n )  [inline]
```

Definition at line 135 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::update_index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator+=
Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

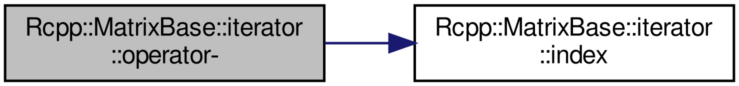
6.348.4.8 operator-() [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
difference_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (
             const iterator & other ) const  [inline]
```

Definition at line 170 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator-
Rcpp::MatrixBase::iterator ::index

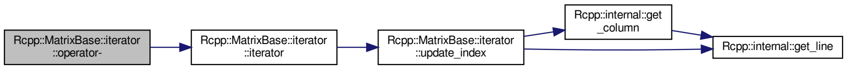
6.348.4.9 operator-() [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (
             difference_type n ) const  [inline]
```

Definition at line 131 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator-
Rcpp::MatrixBase::iterator ::iterator
Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

6.348.4.10 operator--() [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- ( )  [inline]
```

Definition at line 114 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

6.348.4.11 operator--() [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- (
             int  )  [inline]
```

Definition at line 122 of file MatrixBase.h.

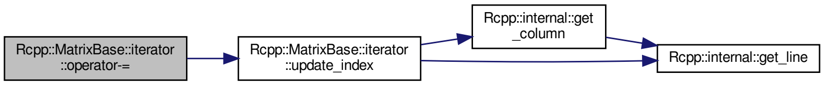
6.348.4.12 operator-=()

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-= (
             difference_type n )  [inline]
```

Definition at line 139 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::update_index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator-=
Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

6.348.4.13 operator->()

```
template<int RTYPE, bool na, typename MATRIX >
pointer Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-> ( )  [inline]
```

Definition at line 147 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object.

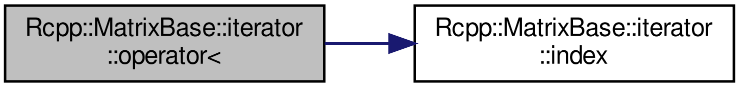
6.348.4.14 operator<()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator< (
             const iterator & other ) const  [inline]
```

Definition at line 157 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator<
Rcpp::MatrixBase::iterator ::index

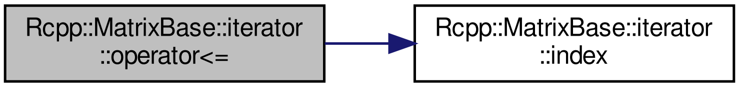
6.348.4.15 operator<=()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<= (
             const iterator & other ) const  [inline]
```

Definition at line 163 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator<=
Rcpp::MatrixBase::iterator ::index

6.348.4.16 operator==()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator== (
            const iterator & y ) const  [inline]
```

Definition at line 151 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↩ ::iterator::j.

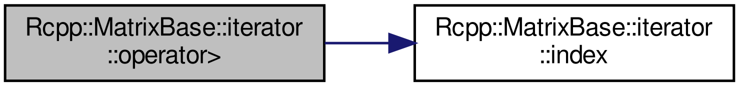
6.348.4.17 operator>()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator> (
            const iterator & other ) const  [inline]
```

Definition at line 160 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator>
Rcpp::MatrixBase::iterator ::index

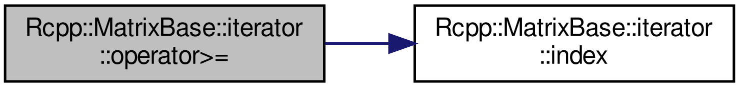
6.348.4.18 operator>=()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>= (
            const iterator & other ) const  [inline]
```

Definition at line 166 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::operator>=
Rcpp::MatrixBase::iterator ::index

6.348.4.19 update_index()

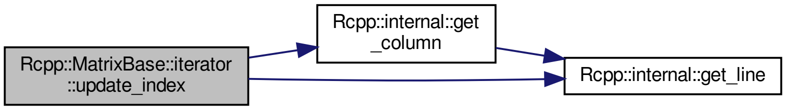
```
template<int RTYPE, bool na, typename MATRIX >
void Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index (
            int index_ )  [inline], [private]
```

Definition at line 180 of file MatrixBase.h.

References Rcpp::internal::get_column(), Rcpp::internal::get_line(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-=().

Here is the call graph for this function:


Rcpp::MatrixBase::iterator ::update_index
Rcpp::internal::get _column
Rcpp::internal::get_line

6.348.5 Member Data Documentation

6.348.5.1 i

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i  [private]
```

Definition at line 177 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator*(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

6.348.5.2 j

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j  [private]
```

Definition at line 177 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator*(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

6.348.5.3 nc

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nc  [private]
```

Definition at line 178 of file MatrixBase.h.

6.348.5.4 nr

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr  [private]
```

Definition at line 178 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp↩ ::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator↩ ::operator-=(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index().

6.348.5.5 object

```
template<int RTYPE, bool na, typename MATRIX >
const MatrixBase& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object  [private]
```

Definition at line 176 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator*(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->().

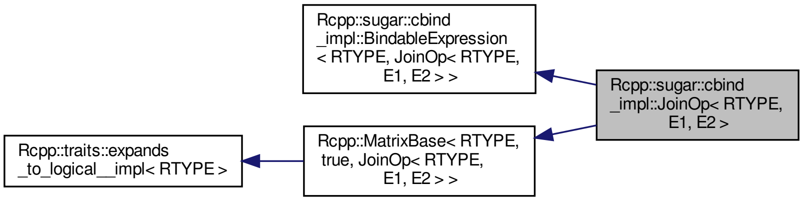
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

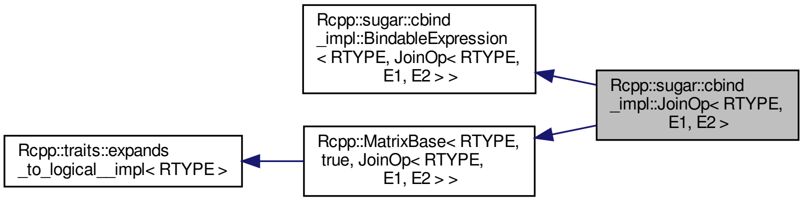
6.349 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >:


Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, JoinOp< RTYPE, E1, E2 > >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >

Collaboration diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >:



Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type

Public Member Functions

- JoinOp (const BindableExpression< RTYPE, E1 > &e1_, const BindableExpression< RTYPE, E2 > &e2_)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- const E1 & e1
- const E2 & e2

Additional Inherited Members

6.349.1 Detailed Description

template<int RTYPE, typename E1, typename E2>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >

Definition at line 166 of file cbind.h.

6.349.2 Member Typedef Documentation

6.349.2.1 stored_type

```
template<int RTYPE, typename E1 , typename E2 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::stored_type
```

Definition at line 170 of file cbind.h.

6.349.3 Constructor & Destructor Documentation

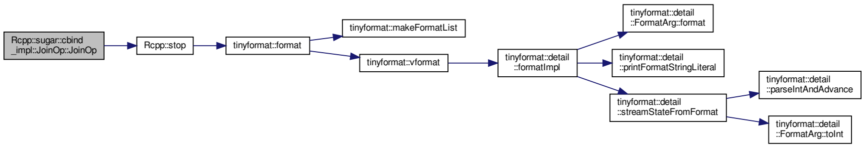
6.349.3.1 JoinOp()

```
template<int RTYPE, typename E1 , typename E2 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::JoinOp (
             const BindableExpression< RTYPE, E1 > & e1_,
             const BindableExpression< RTYPE, E2 > & e2_ )  [inline]
```

Definition at line 177 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2, and Rcpp::stop().

Here is the call graph for this function:



6.349.4 Member Function Documentation

6.349.4.1 ncol()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::ncol ( ) const  [inline]
```

Definition at line 195 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.349.4.2 nrow()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow ( ) const  [inline]
```

Definition at line 193 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator()(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator()(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator()(), and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator()().

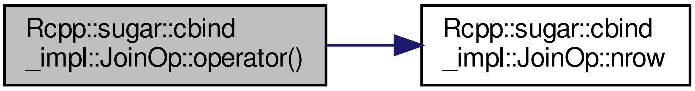
6.349.4.3 operator()()

```
template<int RTYPE, typename E1 , typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator() (
             R_xlen_t i,
             R_xlen_t j ) const  [inline]
```

Definition at line 201 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



6.349.4.4 operator[]()

```
template<int RTYPE, typename E1 , typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 197 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.349.4.5 size()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::size ( ) const  [inline]
```

Definition at line 191 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.349.5 Member Data Documentation

6.349.5.1 e1

```
template<int RTYPE, typename E1 , typename E2 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1  [private]
```

Definition at line 173 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::ncol(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::ncol(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow(), Rcpp::sugar::cbind_impl::Join↩Op< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::nrow(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind_impl↩::JoinOp< RTYPE, E1, E2 >::size(), and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::size().

6.349.5.2 e2

```
template<int RTYPE, typename E1 , typename E2 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2  [private]
```

Definition at line 174 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::ncol(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::ncol(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::ncol(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::nrow(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::nrow(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp<

RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::size(), Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::size(), and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::size().

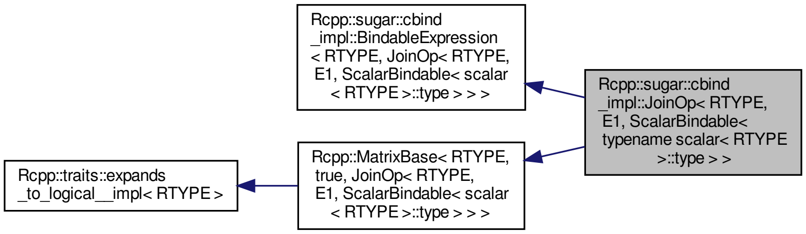
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

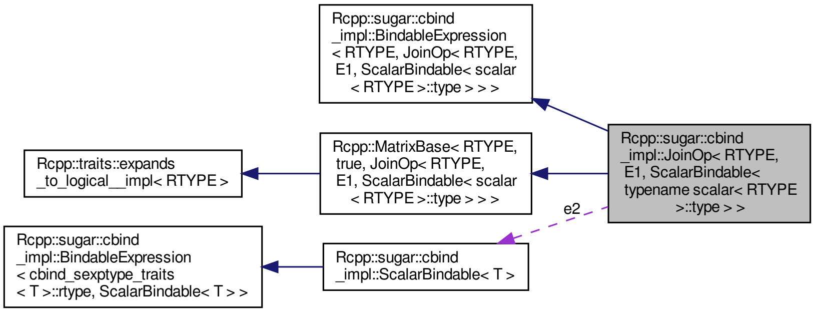
6.350 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, E1, ScalarBindable< scalar < RTYPE >::type > > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, ScalarBindable< scalar < RTYPE >::type > > >

Collaboration diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, E1, ScalarBindable< scalar < RTYPE >::type > > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, ScalarBindable< scalar < RTYPE >::type > > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >
e2
Rcpp::sugar::cbind _impl::BindableExpression < cbind_sexptype_traits < T >::rtype, ScalarBindable< T > >
Rcpp::sugar::cbind _impl::ScalarBindable< T >

Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type
- typedef ScalarBindable< typename scalar< RTYPE >::type > E2

Public Member Functions

- JoinOp (const BindableExpression< RTYPE, E1 > &e1_, const BindableExpression< RTYPE, E2 > &e2_)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- const E1 & e1
- const E2 & e2

Additional Inherited Members

6.350.1 Detailed Description

template<int RTYPE, typename E1>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >

Definition at line 209 of file cbind.h.

6.350.2 Member Typedef Documentation

6.350.2.1 E2

```
template<int RTYPE, typename E1 >
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1,
ScalarBindable< typename scalar< RTYPE >::type > >::E2
```

Definition at line 218 of file cbind.h.

6.350.2.2 stored_type

```
template<int RTYPE, typename E1 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable<
typename scalar< RTYPE >::type > >::stored_type
```

Definition at line 217 of file cbind.h.

6.350.3 Constructor & Destructor Documentation

6.350.3.1 JoinOp()

```
template<int RTYPE, typename E1 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >↩
::JoinOp (
             const BindableExpression< RTYPE, E1 > & e1_,
             const BindableExpression< RTYPE, E2 > & e2_ )  [inline]
```

Definition at line 225 of file cbind.h.

6.350.4 Member Function Documentation

6.350.4.1 ncol()

```
template<int RTYPE, typename E1 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↩
::type > >::ncol ( ) const  [inline]
```

Definition at line 234 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1.

6.350.4.2 nrow()

```
template<int RTYPE, typename E1 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↩
::type > >::nrow ( ) const  [inline]
```

Definition at line 232 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1.

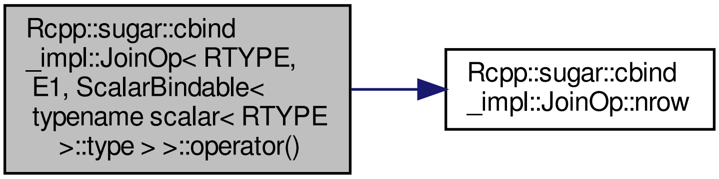
6.350.4.3 operator()()

```
template<int RTYPE, typename E1 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE
>::type > >::operator() (
             R_xlen_t i,
             R_xlen_t j ) const  [inline]
```

Definition at line 240 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



6.350.4.4 operator[]()

```
template<int RTYPE, typename E1 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE
>::type > >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 236 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.350.4.5 size()

```
template<int RTYPE, typename E1 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↩
::type > >::size ( ) const  [inline]
```

Definition at line 230 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1.

6.350.5 Member Data Documentation

6.350.5.1 e1

```
template<int RTYPE, typename E1 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↩
::type > >::e1 [private]
```

Definition at line 221 of file cbind.h.

6.350.5.2 e2

```
template<int RTYPE, typename E1 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↩
::type > >::e2 [private]
```

Definition at line 222 of file cbind.h.

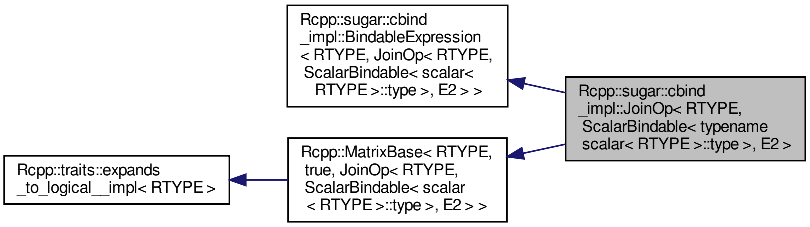
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

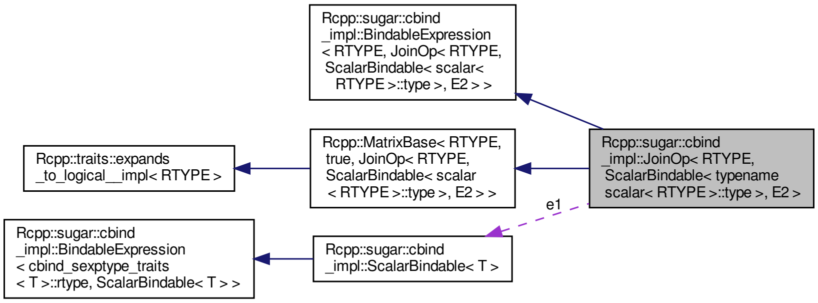
6.351 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↩
::type >, E2 >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar < RTYPE >::type >, E2 > >

Collaboration diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar < RTYPE >::type >, E2 > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >
e1
Rcpp::sugar::cbind _impl::BindableExpression < cbind_sexptype_traits < T >::rtype, ScalarBindable< T > >
Rcpp::sugar::cbind _impl::ScalarBindable< T >

Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type
- typedef ScalarBindable< typename scalar< RTYPE >::type > E1

Public Member Functions

- JoinOp (const BindableExpression< RTYPE, E1 > &e1_, const BindableExpression< RTYPE, E2 > &e2_)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- const E1 & e1
- const E2 & e2

Additional Inherited Members

6.351.1 Detailed Description

template<int RTYPE, typename E2>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >

Definition at line 248 of file cbind.h.

6.351.2 Member Typedef Documentation

6.351.2.1 E1

```
template<int RTYPE, typename E2 >
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, E2 >::E1
```

Definition at line 257 of file cbind.h.

6.351.2.2 stored_type

```
template<int RTYPE, typename E2 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable<
typename scalar< RTYPE >::type >, E2 >::stored_type
```

Definition at line 256 of file cbind.h.

6.351.3 Constructor & Destructor Documentation

6.351.3.1 JoinOp()

```
template<int RTYPE, typename E2 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >↩
::JoinOp (
              const BindableExpression< RTYPE, E1 > & e1_,
              const BindableExpression< RTYPE, E2 > & e2_ )  [inline]
```

Definition at line 264 of file cbind.h.

6.351.4 Member Function Documentation

6.351.4.1 ncol()

```
template<int RTYPE, typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, E2 >::ncol ( ) const  [inline]
```

Definition at line 273 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.351.4.2 nrow()

```
template<int RTYPE, typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, E2 >::nrow ( ) const  [inline]
```

Definition at line 271 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

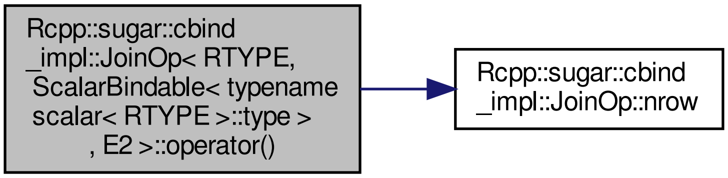
6.351.4.3 operator()()

```
template<int RTYPE, typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↩
::type >, E2 >::operator() (
            R_xlen_t i,
            R_xlen_t j ) const  [inline]
```

Definition at line 279 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:


Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator()
Rcpp::sugar::cbind_impl::JoinOp::nrow

6.351.4.4 operator[]()

```
template<int RTYPE, typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↩
::type >, E2 >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 275 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.351.4.5 size()

```
template<int RTYPE, typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, E2 >::size ( ) const  [inline]
```

Definition at line 269 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.351.5 Member Data Documentation

6.351.5.1 e1

```
template<int RTYPE, typename E2 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, E2 >::e1  [private]
```

Definition at line 260 of file cbind.h.

6.351.5.2 e2

```
template<int RTYPE, typename E2 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, E2 >::e2  [private]
```

Definition at line 261 of file cbind.h.

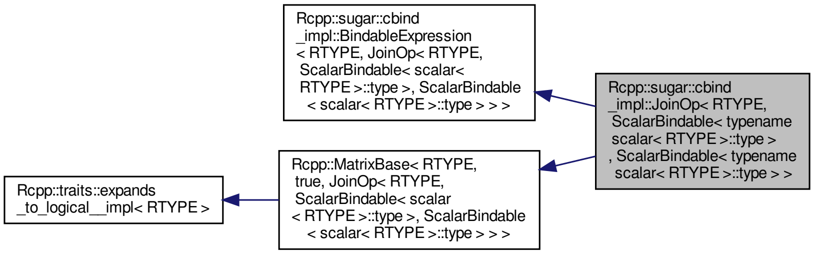
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

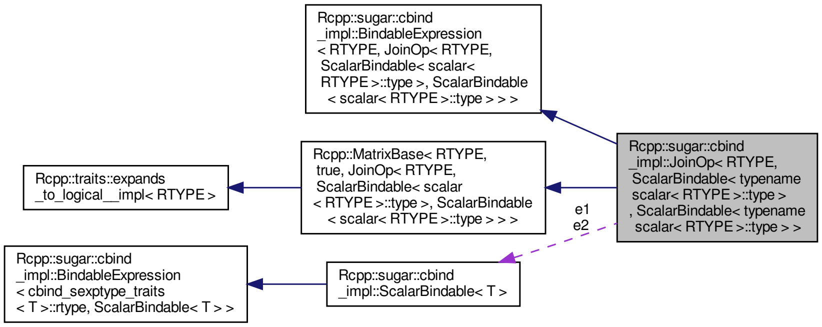
6.352 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, ScalarBindable < scalar< RTYPE >::type > > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > , ScalarBindable< typename scalar< RTYPE >::type > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar < RTYPE >::type >, ScalarBindable < scalar< RTYPE >::type > > >

Collaboration diagram for Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >:


Rcpp::sugar::cbind _impl::BindableExpression < RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, ScalarBindable < scalar< RTYPE >::type > > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar < RTYPE >::type >, ScalarBindable < scalar< RTYPE >::type > > >
Rcpp::sugar::cbind _impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > , ScalarBindable< typename scalar< RTYPE >::type > >
e1
e2
Rcpp::sugar::cbind _impl::BindableExpression < cbind_sexptype_traits < T >::rtype, ScalarBindable< T > >
Rcpp::sugar::cbind _impl::ScalarBindable< T >

Public Types

- typedef cbind_storage_type< RTYPE >::type stored_type
- typedef ScalarBindable< typename scalar< RTYPE >::type > E1
- typedef ScalarBindable< typename scalar< RTYPE >::type > E2

Public Member Functions

- JoinOp (const BindableExpression< RTYPE, E1 > &e1_, const BindableExpression< RTYPE, E2 > &e2_)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- const E1 & e1
- const E2 & e2

Additional Inherited Members

6.352.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >

Definition at line 287 of file cbind.h.

6.352.2 Member Typedef Documentation

6.352.2.1 E1

```
template<int RTYPE>
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >↩
::type > >::E1
```

Definition at line 299 of file cbind.h.

6.352.2.2 E2

```
template<int RTYPE>
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >↩
::type > >::E2
```

Definition at line 300 of file cbind.h.

6.352.2.3 stored_type

```
template<int RTYPE>
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable<
typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::stored_type
```

Definition at line 298 of file cbind.h.

6.352.3 Constructor & Destructor Documentation

6.352.3.1 JoinOp()

```
template<int RTYPE>
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable<
typename scalar< RTYPE >::type > >::JoinOp (
             const BindableExpression< RTYPE, E1 > & e1_,
             const BindableExpression< RTYPE, E2 > & e2_ )  [inline]
```

Definition at line 307 of file cbind.h.

6.352.4 Member Function Documentation

6.352.4.1 ncol()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::ncol ( ) const  [inline]
```

Definition at line 316 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.352.4.2 nrow()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::nrow ( ) const  [inline]
```

Definition at line 314 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

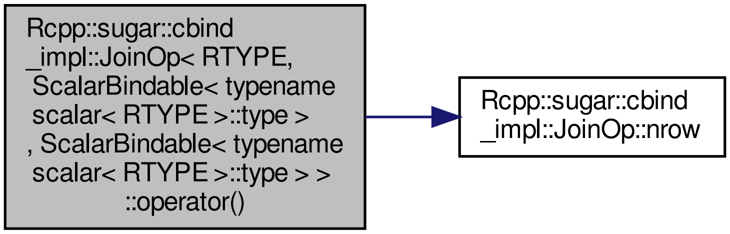
6.352.4.3 operator()()

```
template<int RTYPE>
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↩
::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator() (
            R_xlen_t i,
            R_xlen_t j ) const [inline]
```

Definition at line 322 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:


Rcpp::sugar::cbind _impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > , ScalarBindable< typename scalar< RTYPE >::type > > ::operator()
Rcpp::sugar::cbind _impl::JoinOp::nrow

6.352.4.4 operator[]()

```
template<int RTYPE>
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↩
::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 318 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.352.4.5 size()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::size ( ) const [inline]
```

Definition at line 312 of file cbind.h.

References Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2.

6.352.5 Member Data Documentation

6.352.5.1 e1

```
template<int RTYPE>
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::e1  [private]
```

Definition at line 303 of file cbind.h.

6.352.5.2 e2

```
template<int RTYPE>
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::e2  [private]
```

Definition at line 304 of file cbind.h.

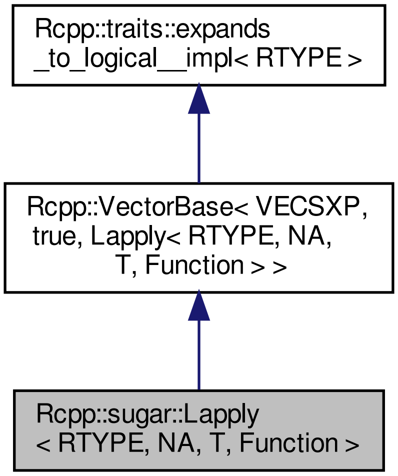
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

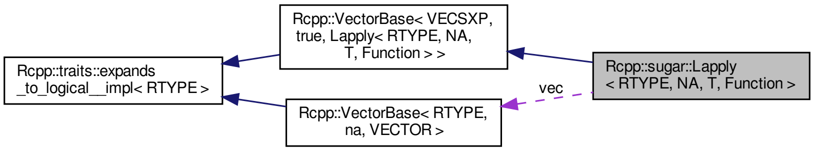
6.353 Rcpp::sugar::Lapply< RTYPE, NA, T, Function > Class Template Reference

```
#include <lapply.h>
```

Inheritance diagram for Rcpp::sugar::Lapply< RTYPE, NA, T, Function >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > >
Rcpp::sugar::Lapply < RTYPE, NA, T, Function >

Collaboration diagram for Rcpp::sugar::Lapply< RTYPE, NA, T, Function >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::sugar::Lapply < RTYPE, NA, T, Function >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC
- typedef ::Rcpp::traits::result_of< Function >::type result_type

Public Member Functions

- Lapply (const VEC &vec_, Function fun_)
- SEXP operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC & vec
- Function fun

6.353.1 Detailed Description

template<int RTYPE, bool NA, typename T, typename Function>
class Rcpp::sugar::Lapply< RTYPE, NA, T, Function >

Definition at line 29 of file lapply.h.

6.353.2 Member Typedef Documentation

6.353.2.1 result_type

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Lapply< RTYPE, NA, T, Function
>::result_type
```

Definition at line 36 of file lapply.h.

6.353.2.2 VEC

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::VEC
```

Definition at line 35 of file lapply.h.

6.353.3 Constructor & Destructor Documentation

6.353.3.1 Lapply()

```
template<int RTYPE, bool NA, typename T , typename Function >
Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::Lapply (
             const VEC & vec_,
             Function fun_ )  [inline]
```

Definition at line 38 of file lapply.h.

6.353.4 Member Function Documentation

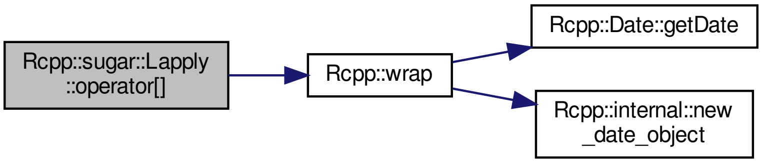
6.353.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function >
SEXP Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 41 of file lapply.h.

References Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::fun, Rcpp::sugar::Lapply< RTYPE, NA, T, Function >↩
::vec, and Rcpp::wrap().

Here is the call graph for this function:



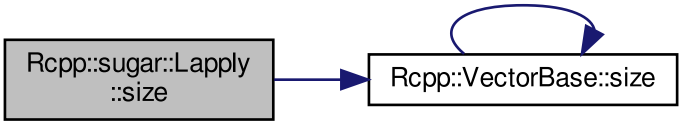
6.353.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function >
R_xlen_t Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size ( ) const  [inline]
```

Definition at line 44 of file lapply.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::vec.

Here is the call graph for this function:



6.353.5 Member Data Documentation

6.353.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function >
Function Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::fun  [private]
```

Definition at line 48 of file lapply.h.

Referenced by Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[ ]().

6.353.5.2 vec

```
template<int RTYPE, bool NA, typename T , typename Function >
const VEC& Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::vec  [private]
```

Definition at line 47 of file lapply.h.

Referenced by Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[ ](), and Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size().

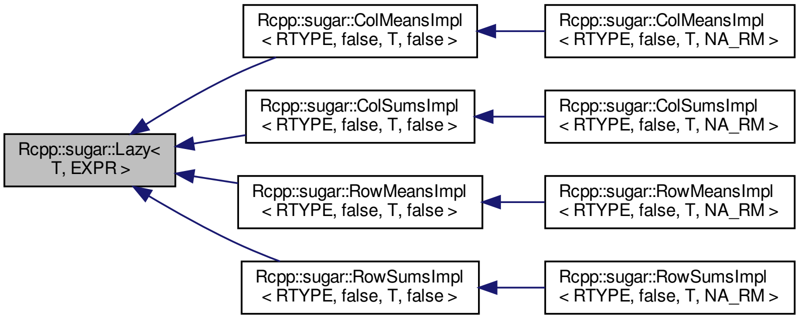
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/lapply.h

6.354 Rcpp::sugar::Lazy< T, EXPR > Class Template Reference

```
#include <Lazy.h>
```

Inheritance diagram for Rcpp::sugar::Lazy< T, EXPR >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::ColMeansImpl < RTYPE, false, T, NA_RM >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::ColSumsImpl < RTYPE, false, T, NA_RM >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, NA_RM >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, NA_RM >

Public Member Functions

- operator T () const

6.354.1 Detailed Description

template<typename T, typename EXPR>
class Rcpp::sugar::Lazy< T, EXPR >

Definition at line 29 of file Lazy.h.

6.354.2 Member Function Documentation

6.354.2.1 operator T()

```
template<typename T , typename EXPR >
Rcpp::sugar::Lazy< T, EXPR >::operator T ( ) const  [inline]
```

Definition at line 31 of file Lazy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/Lazy.h

6.355 Rcpp::internal::LazyVector< VECTOR > Class Template Reference

```
#include <LazyVector.h>
```

Public Types

- typedef VECTOR::r_type r_type
- typedef Rcpp::traits::storage_type< r_type::value >::type stored_type

Public Member Functions

- LazyVector (const VECTOR &vec_)
- stored_type operator[ ] (R_xlen_t i) const

Private Attributes

- const VECTOR & vec
- R_xlen_t n
- std::vector< stored_type > data
- std::vector< bool > known

6.355.1 Detailed Description

template<typename VECTOR>
class Rcpp::internal::LazyVector< VECTOR >

Definition at line 29 of file LazyVector.h.

6.355.2 Member Typedef Documentation

6.355.2.1 r_type

```
template<typename VECTOR >
typedef VECTOR::r_type Rcpp::internal::LazyVector< VECTOR >::r_type
```

Definition at line 31 of file LazyVector.h.

6.355.2.2 stored_type

```
template<typename VECTOR >
typedef Rcpp::traits::storage_type< r_type::value >::type Rcpp::internal::LazyVector< VECTOR >↩
::stored_type
```

Definition at line 32 of file LazyVector.h.

6.355.3 Constructor & Destructor Documentation

6.355.3.1 LazyVector()

```
template<typename VECTOR >
Rcpp::internal::LazyVector< VECTOR >::LazyVector (
            const VECTOR & vec_ )  [inline]
```

Definition at line 34 of file LazyVector.h.

6.355.4 Member Function Documentation

6.355.4.1 operator[]()

```
template<typename VECTOR >
stored_type Rcpp::internal::LazyVector< VECTOR >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 36 of file LazyVector.h.

References Rcpp::internal::LazyVector< VECTOR >::data, Rcpp::internal::LazyVector< VECTOR >::known, and Rcpp::internal::LazyVector< VECTOR >::vec.

6.355.5 Member Data Documentation

6.355.5.1 data

```
template<typename VECTOR >
std::vector<stored_type> Rcpp::internal::LazyVector< VECTOR >::data  [mutable], [private]
```

Definition at line 50 of file LazyVector.h.

Referenced by Rcpp::internal::LazyVector< VECTOR >::operator[ ]().

6.355.5.2 known

```
template<typename VECTOR >
std::vector<bool> Rcpp::internal::LazyVector< VECTOR >::known  [mutable], [private]
```

Definition at line 51 of file LazyVector.h.

Referenced by Rcpp::internal::LazyVector< VECTOR >::operator[ ]().

6.355.5.3 n

```
template<typename VECTOR >
R_xlen_t Rcpp::internal::LazyVector< VECTOR >::n  [private]
```

Definition at line 49 of file LazyVector.h.

6.355.5.4 vec

```
template<typename VECTOR >
const VECTOR& Rcpp::internal::LazyVector< VECTOR >::vec  [private]
```

Definition at line 48 of file LazyVector.h.

Referenced by Rcpp::internal::LazyVector< VECTOR >::operator[ ](), and Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::operator[ ]().

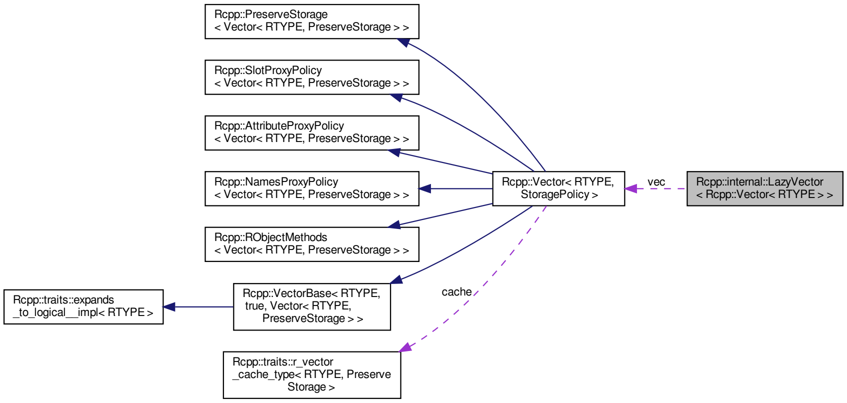
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

6.356 Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > > Class Template Reference

```
#include <LazyVector.h>
```

Collaboration diagram for Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
Rcpp::traits::r_vector
_cache_type< RTYPE, Preserve
Storage >
Rcpp::Vector< RTYPE,
StoragePolicy >
vec
cache
Rcpp::internal::LazyVector
< Rcpp::Vector< RTYPE > >

Public Types

- typedef Rcpp::Vector< RTYPE > VECTOR
- typedef VECTOR::Proxy Proxy

Public Member Functions

- LazyVector (const VECTOR &vec_)
- Proxy operator[ ] (R_xlen_t i) const

Private Attributes

- const VECTOR & vec

6.356.1 Detailed Description

template< int RTYPE >
class Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >

Definition at line 55 of file LazyVector.h.

6.356.2 Member Typedef Documentation

6.356.2.1 Proxy

```
template<int RTYPE>
typedef VECTOR::Proxy Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::Proxy
```

Definition at line 58 of file LazyVector.h.

6.356.2.2 VECTOR

```
template<int RTYPE>
typedef Rcpp::Vector<RTYPE> Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::VECTOR
```

Definition at line 57 of file LazyVector.h.

6.356.3 Constructor & Destructor Documentation

6.356.3.1 LazyVector()

```
template<int RTYPE>
Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::LazyVector (
             const VECTOR & vec_ )  [inline]
```

Definition at line 60 of file LazyVector.h.

6.356.4 Member Function Documentation

6.356.4.1 operator[]()

```
template<int RTYPE>
Proxy Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 61 of file LazyVector.h.

References Rcpp::internal::LazyVector< VECTOR >::vec.

6.356.5 Member Data Documentation

6.356.5.1 vec

```
template<int RTYPE>
const VECTOR& Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::vec  [private]
```

Definition at line 64 of file LazyVector.h.

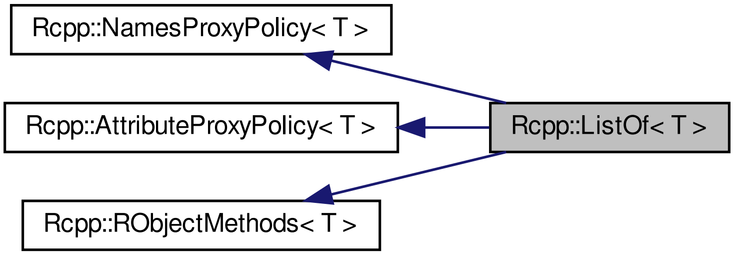
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

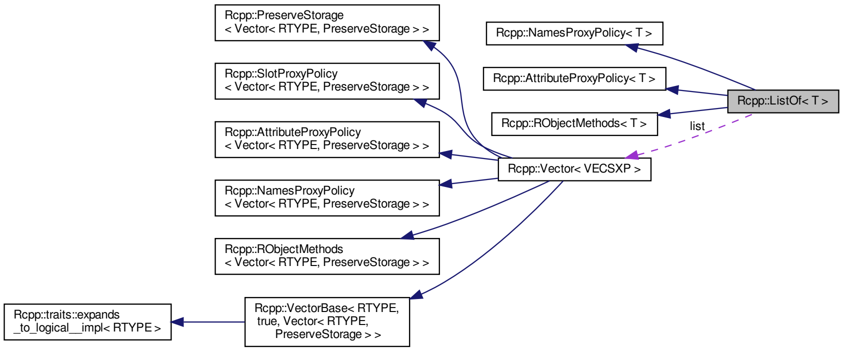
6.357 Rcpp::ListOf< T > Class Template Reference

```
#include <ListOf.h>
```

Inheritance diagram for Rcpp::ListOf< T >:


Rcpp::NamesProxyPolicy< T >
Rcpp::AttributeProxyPolicy< T >
Rcpp::ListOf< T >
Rcpp::RObjectMethods< T >

Collaboration diagram for Rcpp::ListOf< T >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy< T >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy< T >
Rcpp::ListOf< T >
Rcpp::RObjectMethods< T >
list
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< VECSXP >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >

Public Types

- typedef traits::r_vector_iterator< VECSXP >::type iterator
- typedef traits::r_vector_const_iterator< VECSXP >::type const_iterator

Public Member Functions

- ListOf ()
- ListOf (SEXP data_)

- template<typename U >
  ListOf (const U &data_)
- ListOf (const ListOf &other)
- ListOf & operator= (const ListOf &other)
- template<typename U >
  ListOf & operator= (const U &other)
- ChildVector< T > operator[ ] (R_xlen_t i)
- const ChildVector< T > operator[ ] (R_xlen_t i) const
- ChildVector< T > operator[ ] (const std::string &str)
- const ChildVector< T > operator[ ] (const std::string &str) const
- iterator begin ()
- iterator end ()
- const_iterator begin () const
- const_iterator end () const
- R_xlen_t size () const
- List get () const
- operator SEXP () const
- operator List () const

Private Attributes

- List list

6.357.1 Detailed Description

template<typename T>
class Rcpp::ListOf< T >

Definition at line 26 of file ListOf.h.

6.357.2 Member Typedef Documentation

6.357.2.1 const_iterator

```
template<typename T >
typedef traits::r_vector_const_iterator<VECSXP>::type Rcpp::ListOf< T >::const_iterator
```

Definition at line 34 of file ListOf.h.

6.357.2.2 iterator

```
template<typename T >
typedef traits::r_vector_iterator<VECSXP>::type Rcpp::ListOf< T >::iterator
```

Definition at line 33 of file ListOf.h.

6.357.3 Constructor & Destructor Documentation

6.357.3.1 ListOf() [1/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf ( )  [inline]
```

Definition at line 36 of file ListOf.h.

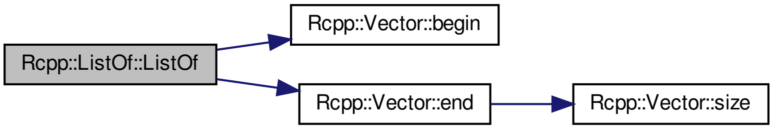
6.357.3.2 ListOf() [2/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf (
              SEXP data_ )  [inline]
```

Definition at line 38 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



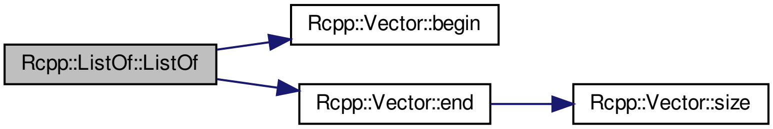
6.357.3.3 ListOf() [3/4]

```
template<typename T >
template<typename U >
Rcpp::ListOf< T >::ListOf (
            const U & data_ )  [inline]
```

Definition at line 43 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



6.357.3.4 ListOf() [4/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf (
            const ListOf< T > & other )  [inline]
```

Definition at line 47 of file ListOf.h.

6.357.4 Member Function Documentation

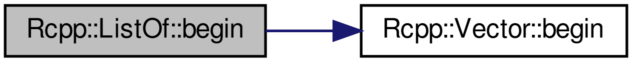
6.357.4.1 begin() [1/2]

```
template<typename T >
iterator Rcpp::ListOf< T >::begin ( ) [inline]
```

Definition at line 82 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



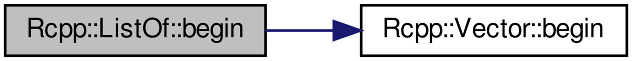
6.357.4.2 begin() [2/2]

```
template<typename T >
const_iterator Rcpp::ListOf< T >::begin ( ) const [inline]
```

Definition at line 90 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



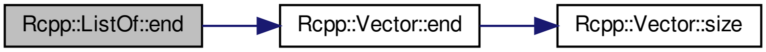
6.357.4.3 end() [1/2]

```
template<typename T >
iterator Rcpp::ListOf< T >::end ( ) [inline]
```

Definition at line 86 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



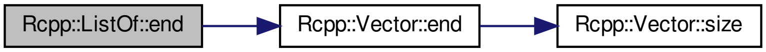
6.357.4.4 end() [2/2]

```
template<typename T >
const_iterator Rcpp::ListOf< T >::end ( ) const [inline]
```

Definition at line 94 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



6.357.4.5 get()

```
template<typename T >
List Rcpp::ListOf< T >::get ( ) const [inline]
```

Definition at line 102 of file ListOf.h.

References Rcpp::ListOf< T >::list.

Referenced by Rcpp::lapply(), and Rcpp::sapply().

6.357.4.6 operator List()

```
template<typename T >
Rcpp::ListOf< T >::operator List ( ) const  [inline]
```

Definition at line 108 of file ListOf.h.

References Rcpp::ListOf< T >::list.

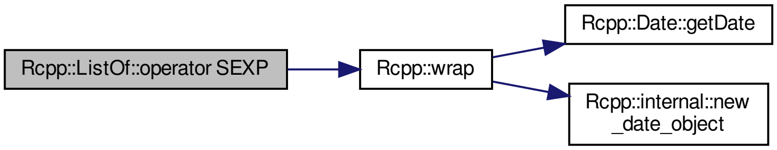
6.357.4.7 operator SEXP()

```
template<typename T >
Rcpp::ListOf< T >::operator SEXP ( ) const  [inline]
```

Definition at line 107 of file ListOf.h.

References Rcpp::ListOf< T >::list, and Rcpp::wrap().

Here is the call graph for this function:



6.357.4.8 operator=() [1/2]

```
template<typename T >
ListOf& Rcpp::ListOf< T >::operator= (
            const ListOf< T > & other )  [inline]
```

Definition at line 49 of file ListOf.h.

References Rcpp::ListOf< T >::list.

6.357.4.9 operator=() [2/2]

```
template<typename T >
template<typename U >
ListOf& Rcpp::ListOf< T >::operator= (
            const U & other ) [inline]
```

Definition at line 57 of file ListOf.h.

References Rcpp::ListOf< T >::list.

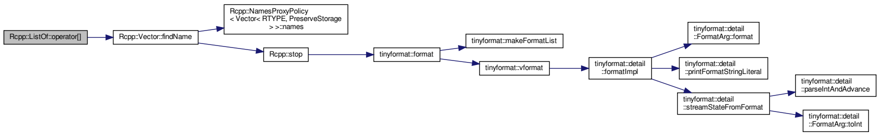
6.357.4.10 operator[]() [1/4]

```
template<typename T >
ChildVector<T> Rcpp::ListOf< T >::operator[] (
            const std::string & str ) [inline]
```

Definition at line 72 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::findName(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:


Rcpp::ListOf::operator[]
Rcpp::Vector::findName
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

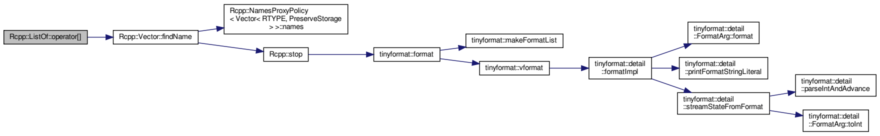
6.357.4.11 operator[]() [2/4]

```
template<typename T >
const ChildVector<T> Rcpp::ListOf< T >::operator[] (
            const std::string & str ) const [inline]
```

Definition at line 76 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::findName(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:


Rcpp::ListOf::operator[]
Rcpp::Vector::findName
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.357.4.12 operator[]() [3/4]

```
template<typename T >
ChildVector<T> Rcpp::ListOf< T >::operator[] (
             R_xlen_t i ) [inline]
```

Definition at line 64 of file ListOf.h.

References Rcpp::ListOf< T >::list.

6.357.4.13 operator[]() [4/4]

```
template<typename T >
const ChildVector<T> Rcpp::ListOf< T >::operator[] (
             R_xlen_t i ) const [inline]
```

Definition at line 68 of file ListOf.h.

References Rcpp::ListOf< T >::list.

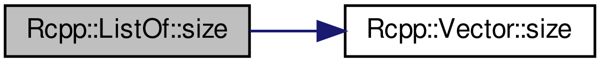
6.357.4.14 size()

```
template<typename T >
R_xlen_t Rcpp::ListOf< T >::size ( ) const [inline]
```

Definition at line 98 of file ListOf.h.

References Rcpp::ListOf< T >::list, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.357.5 Member Data Documentation

6.357.5.1 list

```
template<typename T >
List Rcpp::ListOf< T >::list  [private]
```

Definition at line 112 of file ListOf.h.

Referenced by Rcpp::ListOf< T >::begin(), Rcpp::ListOf< T >::end(), Rcpp::ListOf< T >::get(), Rcpp::ListOf< T >↩ ::ListOf(), Rcpp::ListOf< T >::operator List(), Rcpp::ListOf< T >::operator SEXP(), Rcpp::ListOf< T >::operator=(), Rcpp::ListOf< T >::operator[ ](), and Rcpp::ListOf< T >::size().

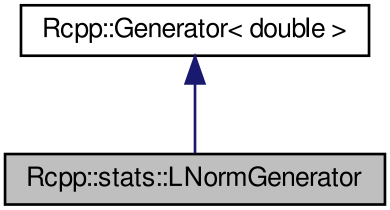
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ListOf.h

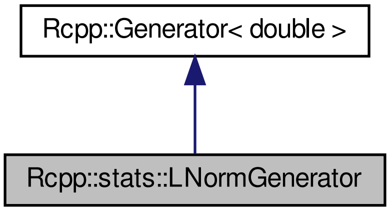
6.358 Rcpp::stats::LNormGenerator Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for Rcpp::stats::LNormGenerator:



Collaboration diagram for Rcpp::stats::LNormGenerator:



Public Member Functions

- LNormGenerator (double meanlog_=0.0, double sdlog_=1.0)
- double operator() () const

Private Attributes

- double meanlog
- double sdlog

Additional Inherited Members

6.358.1 Detailed Description

Definition at line 28 of file rlnorm.h.

6.358.2 Constructor & Destructor Documentation

6.358.2.1 LNormGenerator()

```
Rcpp::stats::LNormGenerator::LNormGenerator (
             double meanlog_ = 0.0,
             double sdlog_ = 1.0 )  [inline]
```

Definition at line 31 of file rlnorm.h.

6.358.3 Member Function Documentation

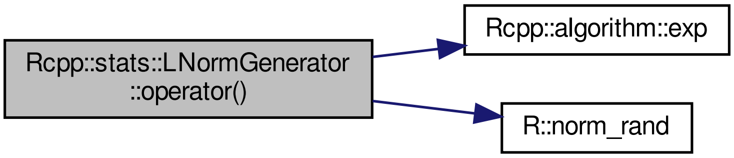
6.358.3.1 operator()()

```
double Rcpp::stats::LNormGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rlnorm.h.

References Rcpp::algorithm::exp(), meanlog, R::norm_rand(), and sdlog.

Here is the call graph for this function:



6.358.4 Member Data Documentation

6.358.4.1 meanlog

```
double Rcpp::stats::LNormGenerator::meanlog  [private]
```

Definition at line 39 of file rlnorm.h.

Referenced by operator()().

6.358.4.2 sdlog

```
double Rcpp::stats::LNormGenerator::sdlog  [private]
```

Definition at line 40 of file rlnorm.h.

Referenced by operator()().

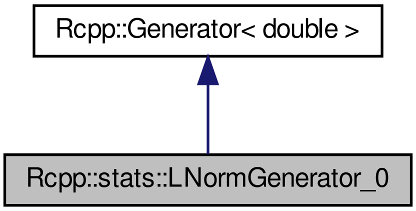
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

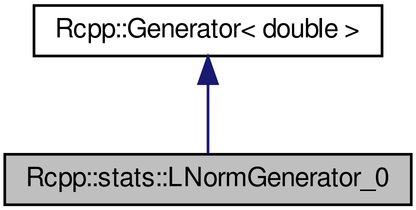
6.359 Rcpp::stats::LNormGenerator_0 Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for Rcpp::stats::LNormGenerator_0:



Collaboration diagram for Rcpp::stats::LNormGenerator_0:



Public Member Functions

- LNormGenerator_0 ()
- double operator() () const

Additional Inherited Members

6.359.1 Detailed Description

Definition at line 59 of file rlnorm.h.

6.359.2 Constructor & Destructor Documentation

6.359.2.1 LNormGenerator_0()

```
Rcpp::stats::LNormGenerator_0::LNormGenerator_0 ( )  [inline]
```

Definition at line 62 of file rlnorm.h.

6.359.3 Member Function Documentation

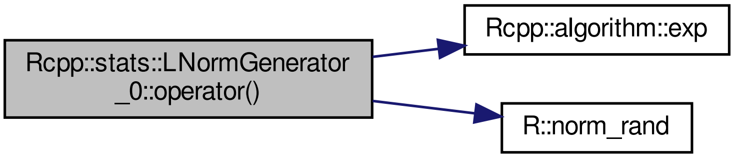
6.359.3.1 operator()()

```
double Rcpp::stats::LNormGenerator_0::operator() ( ) const [inline]
```

Definition at line 64 of file rlnorm.h.

References Rcpp::algorithm::exp(), and R::norm_rand().

Here is the call graph for this function:



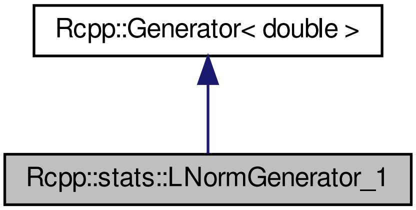
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

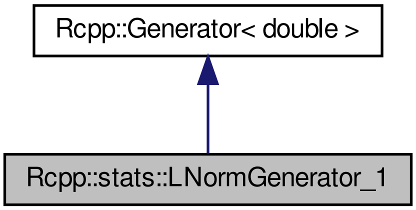
6.360 Rcpp::stats::LNormGenerator_1 Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for Rcpp::stats::LNormGenerator_1:



Collaboration diagram for Rcpp::stats::LNormGenerator_1:



Public Member Functions

- LNormGenerator_1 (double meanlog_=0.0)
- double operator() () const

Private Attributes

- double meanlog

Additional Inherited Members

6.360.1 Detailed Description

Definition at line 44 of file rlnorm.h.

6.360.2 Constructor & Destructor Documentation

6.360.2.1 LNormGenerator_1()

```
Rcpp::stats::LNormGenerator_1::LNormGenerator_1 (
             double meanlog_ = 0.0 )  [inline]
```

Definition at line 47 of file rlnorm.h.

6.360.3 Member Function Documentation

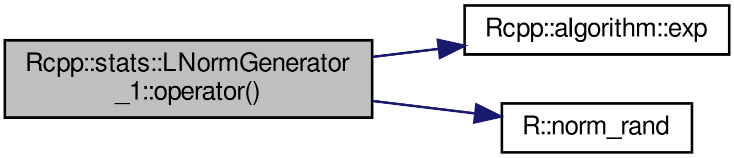
6.360.3.1 operator()()

```
double Rcpp::stats::LNormGenerator_1::operator() ( ) const  [inline]
```

Definition at line 50 of file rlnorm.h.

References Rcpp::algorithm::exp(), meanlog, and R::norm_rand().

Here is the call graph for this function:



6.360.4 Member Data Documentation

6.360.4.1 meanlog

```
double Rcpp::stats::LNormGenerator_1::meanlog  [private]
```

Definition at line 55 of file rlnorm.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlnorm.h

6.361 Rcpp::algorithm::helpers::log Struct Reference

```
#include <algorithm.h>
```

Public Member Functions

- template< typename T >
  double operator() (T val)

6.361.1 Detailed Description

Definition at line 195 of file algorithm.h.

6.361.2 Member Function Documentation

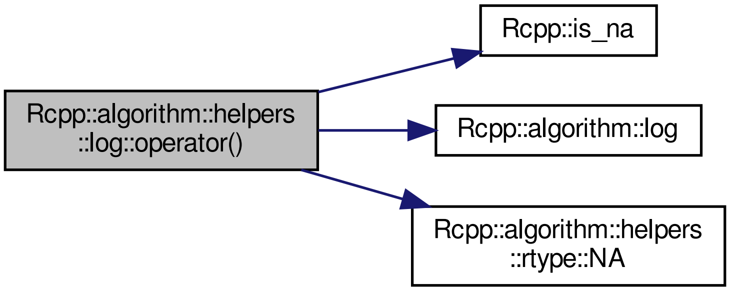
6.361.2.1 operator()()

```
template<typename T >
double Rcpp::algorithm::helpers::log::operator() (
            T val ) [inline]
```

Definition at line 197 of file algorithm.h.

References Rcpp::is_na(), Rcpp::algorithm::log(), and Rcpp::algorithm::helpers::rtype< T >::NA().

Here is the call graph for this function:


Rcpp::is_na
Rcpp::algorithm::helpers ::log::operator()
Rcpp::algorithm::log
Rcpp::algorithm::helpers ::rtype::NA

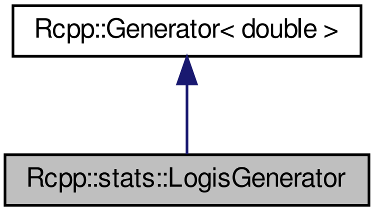
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

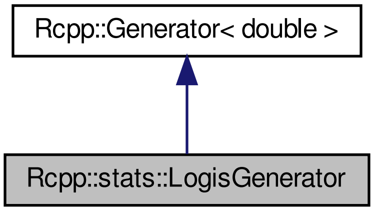
6.362 Rcpp::stats::LogisGenerator Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for Rcpp::stats::LogisGenerator:



Collaboration diagram for Rcpp::stats::LogisGenerator:



Public Member Functions

- LogisGenerator (double location_, double scale_)
- double operator() () const

Private Attributes

- double location
- double scale

Additional Inherited Members

6.362.1 Detailed Description

Definition at line 28 of file rlogis.h.

6.362.2 Constructor & Destructor Documentation

6.362.2.1 LogisGenerator()

```
Rcpp::stats::LogisGenerator::LogisGenerator (
            double location_,
            double scale_ )  [inline]
```

Definition at line 31 of file rlogis.h.

6.362.3 Member Function Documentation

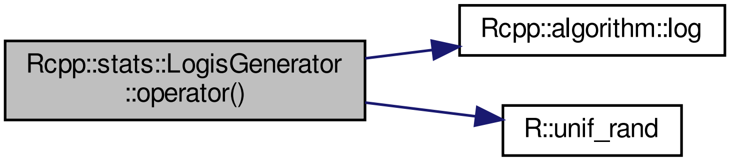
6.362.3.1 operator()()

```
double Rcpp::stats::LogisGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rlogis.h.

References location, Rcpp::algorithm::log(), scale, and R::unif_rand().

Here is the call graph for this function:



6.362.4 Member Data Documentation

6.362.4.1 location

```
double Rcpp::stats::LogisGenerator::location  [private]
```

Definition at line 40 of file rlogis.h.

Referenced by operator()().

6.362.4.2 scale

```
double Rcpp::stats::LogisGenerator::scale  [private]
```

Definition at line 41 of file rlogis.h.

Referenced by operator()().

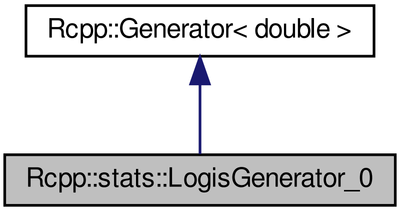
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

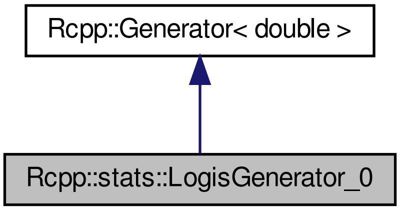
6.363 Rcpp::stats::LogisGenerator_0 Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for Rcpp::stats::LogisGenerator_0:



Collaboration diagram for Rcpp::stats::LogisGenerator_0:



Public Member Functions

- LogisGenerator_0 ()
- double operator() () const

Additional Inherited Members

6.363.1 Detailed Description

Definition at line 61 of file rlogis.h.

6.363.2 Constructor & Destructor Documentation

6.363.2.1 LogisGenerator_0()

```
Rcpp::stats::LogisGenerator_0::LogisGenerator_0 ( )  [inline]
```

Definition at line 64 of file rlogis.h.

6.363.3 Member Function Documentation

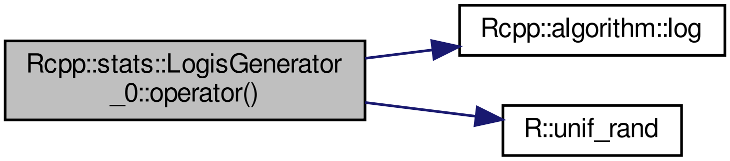
6.363.3.1 operator()()

```
double Rcpp::stats::LogisGenerator_0::operator() ( ) const  [inline]
```

Definition at line 66 of file rlogis.h.

References Rcpp::algorithm::log(), and R::unif_rand().

Here is the call graph for this function:



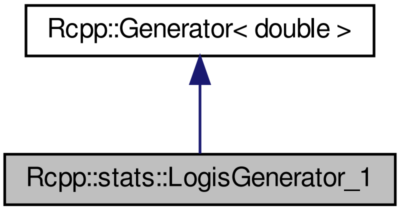
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

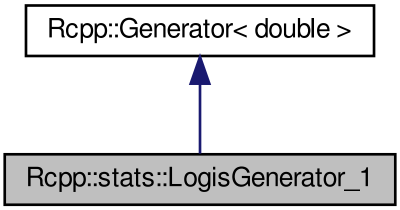
6.364 Rcpp::stats::LogisGenerator_1 Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for Rcpp::stats::LogisGenerator_1:



Collaboration diagram for Rcpp::stats::LogisGenerator_1:



Public Member Functions

- LogisGenerator_1 (double location_)
- double operator() () const

Private Attributes

- double location

Additional Inherited Members

6.364.1 Detailed Description

Definition at line 45 of file rlogis.h.

6.364.2 Constructor & Destructor Documentation

6.364.2.1 LogisGenerator_1()

```
Rcpp::stats::LogisGenerator_1::LogisGenerator_1 (
            double location_ )  [inline]
```

Definition at line 48 of file rlogis.h.

6.364.3 Member Function Documentation

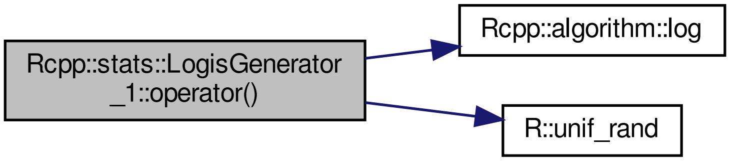
6.364.3.1 operator()()

```
double Rcpp::stats::LogisGenerator_1::operator() ( ) const  [inline]
```

Definition at line 51 of file rlogis.h.

References location, Rcpp::algorithm::log(), and R::unif_rand().

Here is the call graph for this function:



6.364.4 Member Data Documentation

6.364.4.1 location

```
double Rcpp::stats::LogisGenerator_1::location  [private]
```

Definition at line 57 of file rlogis.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rlogis.h

6.365 Rcpp::LongjumpException Struct Reference

```
#include <exceptions.h>
```

Public Member Functions

- LongjumpException (SEXP token_)

Public Attributes

- SEXP token

6.365.1 Detailed Description

Definition at line 164 of file exceptions.h.

6.365.2 Constructor & Destructor Documentation

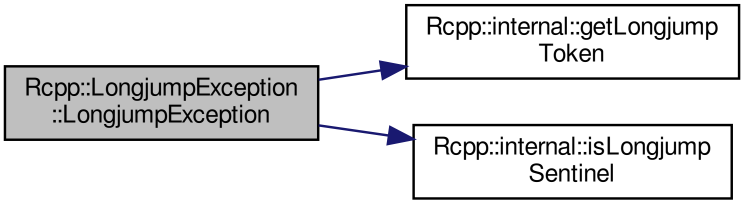
6.365.2.1 LongjumpException()

```
Rcpp::LongjumpException::LongjumpException (
             SEXP token_ )  [inline]
```

Definition at line 166 of file exceptions.h.

References Rcpp::internal::getLongjumpToken(), Rcpp::internal::isLongjumpSentinel(), and token.

Here is the call graph for this function:



6.365.3 Member Data Documentation

6.365.3.1 token

```
SEXP Rcpp::LongjumpException::token
```

Definition at line 165 of file exceptions.h.

Referenced by LongjumpException().

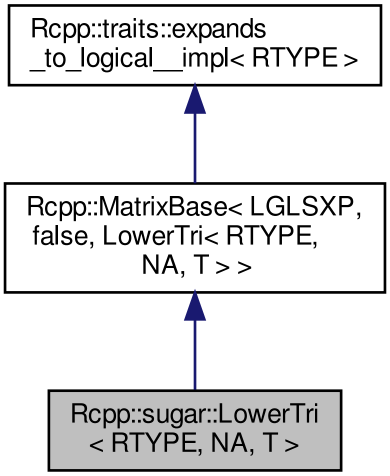
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/exceptions.h

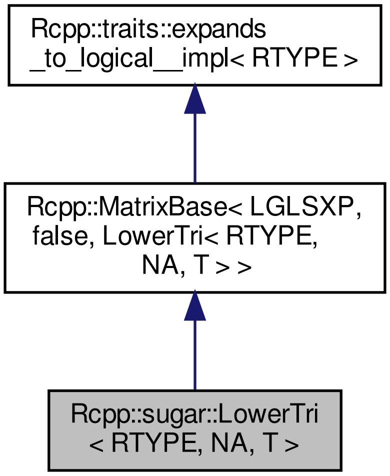
6.366 Rcpp::sugar::LowerTri< RTYPE, NA, T > Class Template Reference

```
#include <lower_tri.h>
```

Inheritance diagram for Rcpp::sugar::LowerTri< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::LowerTri< RTYPE, NA, T >:



Public Types

- typedef Rcpp::MatrixBase< RTYPE, NA, T > MatBase

Public Member Functions

- LowerTri (const T &lhs, bool diag)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Types

- typedef bool(LowerTri::∗ Method) (int, int) const

Private Member Functions

- bool get_diag_true (int i, int j) const
- bool get_diag_false (int i, int j) const
- bool get (int i, int j) const

Private Attributes

- int nr
- int nc
- Method getter

Additional Inherited Members

6.366.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::LowerTri< RTYPE, NA, T >

Definition at line 30 of file lower_tri.h.

6.366.2 Member Typedef Documentation

6.366.2.1 MatBase

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::MatrixBase<RTYPE, NA, T> Rcpp::sugar::LowerTri< RTYPE, NA, T >::MatBase
```

Definition at line 32 of file lower_tri.h.

6.366.2.2 Method

```
template<int RTYPE, bool NA, typename T >
typedef bool(LowerTri::* Rcpp::sugar::LowerTri< RTYPE, NA, T >::Method) (int, int) const  [private]
```

Definition at line 47 of file lower_tri.h.

6.366.3 Constructor & Destructor Documentation

6.366.3.1 LowerTri()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::LowerTri< RTYPE, NA, T >::LowerTri (
             const T & lhs,
             bool diag )  [inline]
```

Definition at line 34 of file lower_tri.h.

6.366.4 Member Function Documentation

6.366.4.1 get()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 56 of file lower_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::operator()().

6.366.4.2 get_diag_false()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get_diag_false (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 54 of file lower_tri.h.

6.366.4.3 get_diag_true()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get_diag_true (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 52 of file lower_tri.h.

6.366.4.4 ncol()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::ncol ( ) const  [inline]
```

Definition at line 44 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc.

6.366.4.5 nrow()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nrow ( ) const  [inline]
```

Definition at line 43 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr.

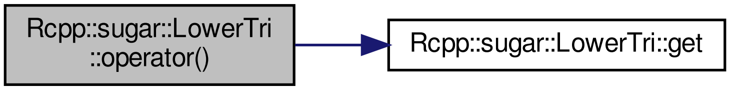
6.366.4.6 operator()()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 40 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::get().

Here is the call graph for this function:


Rcpp::sugar::LowerTri ::operator()
Rcpp::sugar::LowerTri::get

6.366.4.7 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::LowerTri< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 42 of file lower_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc, and Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr.

6.366.5 Member Data Documentation

6.366.5.1 getter

```
template<int RTYPE, bool NA, typename T >
Method Rcpp::sugar::LowerTri< RTYPE, NA, T >::getter  [private]
```

Definition at line 50 of file lower_tri.h.

6.366.5.2 nc

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc  [private]
```

Definition at line 49 of file lower_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::ncol(), and Rcpp::sugar::LowerTri< RTYPE, NA, T >::size().

6.366.5.3 nr

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr  [private]
```

Definition at line 49 of file lower_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::nrow(), and Rcpp::sugar::LowerTri< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/lower_tri.h

6.367 Rcpp::lsinfo Struct Reference

Public Attributes

- time_t ls_trans
- int_fast64_t ls_corr

6.367.1 Detailed Description

Definition at line 388 of file date.cpp.

6.367.2 Member Data Documentation

6.367.2.1 ls_corr

```
int_fast64_t Rcpp::lsinfo::ls_corr
```

Definition at line 390 of file date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

6.367.2.2 ls_trans

```
time_t Rcpp::lsinfo::ls_trans
```

Definition at line 389 of file date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

The documentation for this struct was generated from the following file:

- src/date.cpp

6.368 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container > Struct Template Reference

```
#include <cbind.h>
```

6.368.1 Detailed Description

template<typename T, bool is_container = has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container >

Definition at line 392 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.369 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

Public Types

- enum { RTYPE = cbind_sexptype_traits<stored_type>::rtype }
- typedef Rcpp::Matrix< LGLSXP > T
- typedef bool stored_type

Public Member Functions

- ContainerBindable< LGLSXP, T > operator() (const T &t) const

6.369.1 Detailed Description

Definition at line 409 of file cbind.h.

6.369.2 Member Typedef Documentation

6.369.2.1 stored_type

```
typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >↩
::stored_type
```

Definition at line 411 of file cbind.h.

6.369.2.2 T

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix<
LGLSXP >, true >::T
```

Definition at line 410 of file cbind.h.

6.369.3 Member Enumeration Documentation

6.369.3.1 anonymous enum

```
anonymous enum
```

Enumerator

RTYPE	

Definition at line 413 of file cbind.h.

6.369.4 Member Function Documentation

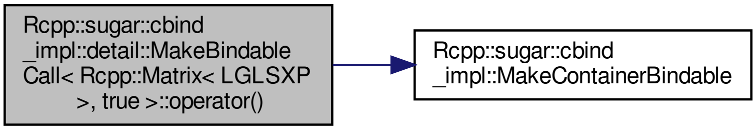
6.369.4.1 operator()()

```
ContainerBindable<LGLSXP, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix<
LGLSXP >, true >::operator() (
            const T & t ) const [inline]
```

Definition at line 415 of file cbind.h.

References Rcpp::sugar::cbind_impl::MakeContainerBindable().

Here is the call graph for this function:


Rcpp::sugar::cbind
_impl::detail::MakeBindable
Call< Rcpp::Matrix< LGLSXP
>, true >::operator()
Rcpp::sugar::cbind
_impl::MakeContainerBindable

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.370 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

Public Types

- enum { RTYPE = cbind_sexptype_traits<stored_type>::rtype }
- typedef Rcpp::Vector< LGLSXP > T
- typedef bool stored_type

Public Member Functions

- ContainerBindable< LGLSXP, T > operator() (const T &t) const

6.370.1 Detailed Description

Definition at line 421 of file cbind.h.

6.370.2 Member Typedef Documentation

6.370.2.1 stored_type

```
typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >↩
::stored_type
```

Definition at line 423 of file cbind.h.

6.370.2.2 T

```
typedef Rcpp::Vector<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector<
LGLSXP >, true >::T
```

Definition at line 422 of file cbind.h.

6.370.3 Member Enumeration Documentation

6.370.3.1 anonymous enum

```
anonymous enum
```

Enumerator

RTYPE	

Definition at line 425 of file cbind.h.

6.370.4 Member Function Documentation

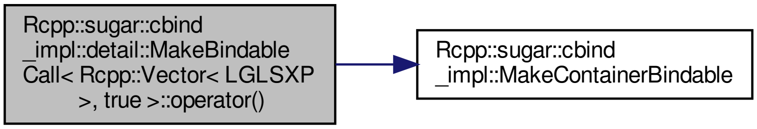
6.370.4.1 operator()()

```
ContainerBindable<LGLSXP, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector<
LGLSXP >, true >::operator() (
             const T & t ) const  [inline]
```

Definition at line 427 of file cbind.h.

References Rcpp::sugar::cbind_impl::MakeContainerBindable().

Here is the call graph for this function:


Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >::operator()
Rcpp::sugar::cbind_impl::MakeContainerBindable

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.371 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false > Struct Template Reference

```
#include <cbind.h>
```

Public Types

- enum { RTYPE = cbind_sexptype_traits<T>::rtype }

Public Member Functions

- ScalarBindable< T > operator() (const T &t) const

6.371.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >

Definition at line 433 of file cbind.h.

6.371.2 Member Enumeration Documentation

6.371.2.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	

Definition at line 434 of file cbind.h.

6.371.3 Member Function Documentation

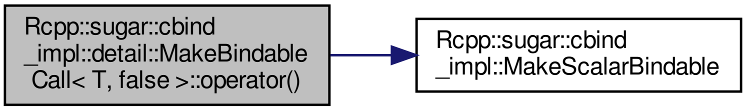
6.371.3.1 operator()()

```
template<typename T >
ScalarBindable<T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >::operator() (
             const T & t ) const  [inline]
```

Definition at line 436 of file cbind.h.

References Rcpp::sugar::cbind_impl::MakeScalarBindable().

Here is the call graph for this function:


Rcpp::sugar::cbind _impl::detail::MakeBindable Call< T, false >::operator()
Rcpp::sugar::cbind _impl::MakeScalarBindable

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.372 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true > Struct Template Reference

```
#include <cbind.h>
```

Public Types

- enum { RTYPE = cbind_sexptype_traits<stored_type>::rtype }
- typedef cbind_storage_type< cbind_sexptype_traits< typename T::stored_type >::rtype >::type stored_type

Public Member Functions

- ContainerBindable< RTYPE, T > operator() (const T &t) const

6.372.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >

Definition at line 395 of file cbind.h.

6.372.2 Member Typedef Documentation

6.372.2.1 stored_type

```
template<typename T >
typedef cbind_storage_type< cbind_sexptype_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind_imp
T, true >::stored_type
```

Definition at line 398 of file cbind.h.

6.372.3 Member Enumeration Documentation

6.372.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	

Definition at line 400 of file cbind.h.

6.372.4 Member Function Documentation

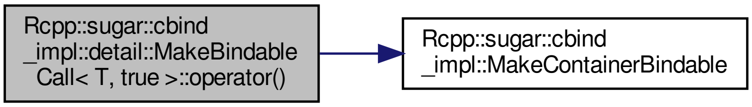
6.372.4.1 operator()()

```
template<typename T >
ContainerBindable<RTYPE, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >::operator()
(
             const T & t ) const  [inline]
```

Definition at line 402 of file cbind.h.

References Rcpp::sugar::cbind_impl::MakeContainerBindable().

Here is the call graph for this function:


Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >::operator()
Rcpp::sugar::cbind_impl::MakeContainerBindable

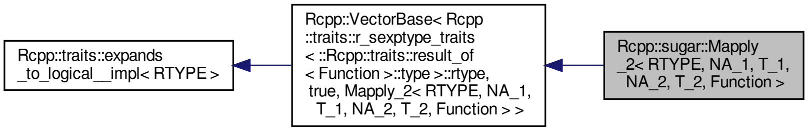
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

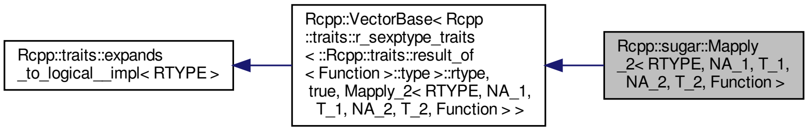
6.373 Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > Class Template Reference

`#include <mapply_2.h>`

Inheritance diagram for Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
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::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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
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::sugar::Mapply
_2< RTYPE, NA_1, T_1,
NA_2, T_2, Function >

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type result_type

Public Member Functions

- Mapply_2 (const T_1 &vec_1_, const T_2 &vec_2_, Function fun_)
- result_type operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T_1 & vec_1
- const T_2 & vec_2
- Function fun

6.373.1 Detailed Description

template<int RTYPE, bool NA_1, typename T_1, bool NA_2, typename T_2, typename Function>
class Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >

Definition at line 33 of file mapply_2.h.

6.373.2 Member Typedef Documentation

6.373.2.1 result_type

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2,
T_2, Function >::result_type
```

Definition at line 41 of file mapply_2.h.

6.373.3 Constructor & Destructor Documentation

6.373.3.1 Mapply_2()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::Mapply_2 (
            const T_1 & vec_1_,
            const T_2 & vec_2_,
            Function fun_ )  [inline]
```

Definition at line 43 of file mapply_2.h.

6.373.4 Member Function Documentation

6.373.4.1 operator[]()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
result_type Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 46 of file mapply_2.h.

References Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun, Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1, and Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_2.

6.373.4.2 size()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::size ( ) const  [inline]
```

Definition at line 49 of file mapply_2.h.

References Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1.

6.373.5 Member Data Documentation

6.373.5.1 fun

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
Function Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun  [private]
```

Definition at line 54 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[ ]().

6.373.5.2 vec_1

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
const T_1& Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1  [private]
```

Definition at line 52 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[ ](), and Rcpp::sugar↩::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::size().

6.373.5.3 vec_2

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
const T_2& Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_2 [private]
```

Definition at line 53 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[ ]().

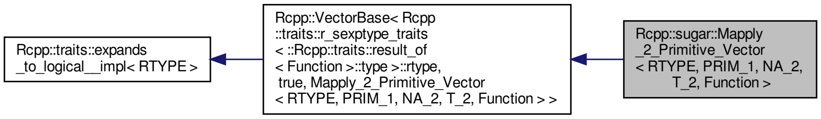
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

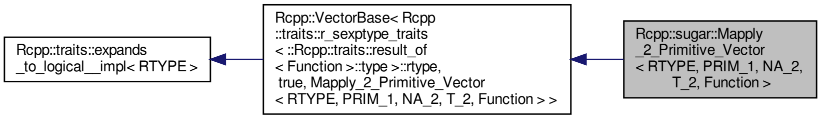
6.374 Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function > Class Template Reference

```
#include <mapply_2.h>
```

Inheritance diagram for Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::sugar::Mapply _2_Primitive_Vector < RTYPE, PRIM_1, NA_2, T_2, Function >

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type result_type

Public Member Functions

- Mapply_2_Primitive_Vector (PRIM_1 prim_1_, const T_2 &vec_2_, Function fun_)
- result_type operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- PRIM_1 prim_1
- const T_2 & vec_2
- Function fun

6.374.1 Detailed Description

template<int RTYPE, typename PRIM_1, bool NA_2, typename T_2, typename Function>
class Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >

Definition at line 93 of file mapply_2.h.

6.374.2 Member Typedef Documentation

6.374.2.1 result_type

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE,
PRIM_1, NA_2, T_2, Function >::result_type
```

Definition at line 103 of file mapply_2.h.

6.374.3 Constructor & Destructor Documentation

6.374.3.1 Mapply_2_Primitive_Vector()

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::Mapply_2_Primitive_Vector
(
            PRIM_1 prim_1_,
            const T_2 & vec_2_,
            Function fun_ )  [inline]
```

Definition at line 105 of file mapply_2.h.

6.374.4 Member Function Documentation

6.374.4.1 operator[]()

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
result_type Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 108 of file mapply_2.h.

References Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::fun, Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::prim_1, and Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::vec_2.

6.374.4.2 size()

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::size ( )
const  [inline]
```

Definition at line 111 of file mapply_2.h.

References Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::vec_2.

6.374.5 Member Data Documentation

6.374.5.1 fun

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
Function Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::fun  [private]
```

Definition at line 116 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[ ]().

6.374.5.2 prim_1

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
PRIM_1 Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::prim_↩
1  [private]
```

Definition at line 114 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[ ]().

6.374.5.3 vec_2

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
const T_2& Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::vec_↩
2  [private]
```

Definition at line 115 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[ ](), and Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::size().

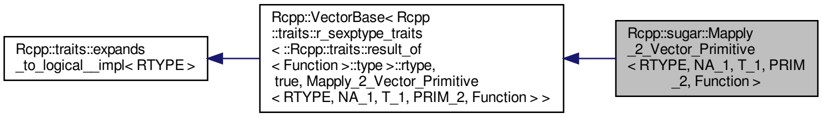
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

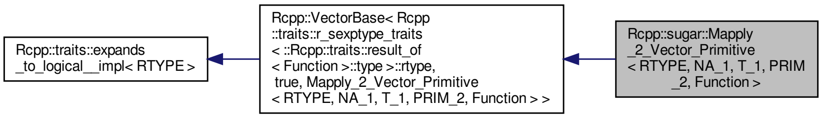
6.375 Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function > Class Template Reference

```
#include <mapply_2.h>
```

Inheritance diagram for Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
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::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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
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::sugar::Mapply
_2_Vector_Primitive
< RTYPE, NA_1, T_1, PRIM
_2, Function >

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type result_type

Public Member Functions

- Mapply_2_Vector_Primitive (const T_1 &vec_1_, PRIM_2 prim_2_, Function fun_)
- result_type operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T_1 & vec_1
- PRIM_2 prim_2
- Function fun

6.375.1 Detailed Description

template<int RTYPE, bool NA_1, typename T_1, typename PRIM_2, typename Function>
class Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >

Definition at line 62 of file mapply_2.h.

6.375.2 Member Typedef Documentation

6.375.2.1 result_type

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE,
NA_1, T_1, PRIM_2, Function >::result_type
```

Definition at line 72 of file mapply_2.h.

6.375.3 Constructor & Destructor Documentation

6.375.3.1 Mapply_2_Vector_Primitive()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::Mapply_2_Vector_Primitive
(
             const T_1 & vec_1_,
             PRIM_2 prim_2_,
             Function fun_ )  [inline]
```

Definition at line 74 of file mapply_2.h.

6.375.4 Member Function Documentation

6.375.4.1 operator[]()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
result_type Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 77 of file mapply_2.h.

References Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::fun, Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::prim_2, and Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::vec_1.

6.375.4.2 size()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::size ( )
const  [inline]
```

Definition at line 80 of file mapply_2.h.

References Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::vec_1.

6.375.5 Member Data Documentation

6.375.5.1 fun

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
Function Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::fun [private]
```

Definition at line 85 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[ ]().

6.375.5.2 prim_2

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
PRIM_2 Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::prim_↩
2 [private]
```

Definition at line 84 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[ ]().

6.375.5.3 vec_1

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
const T_1& Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::vec_↩
1 [private]
```

Definition at line 83 of file mapply_2.h.

Referenced by Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[ ](), and Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::size().

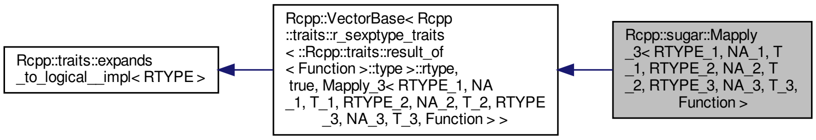
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_2.h

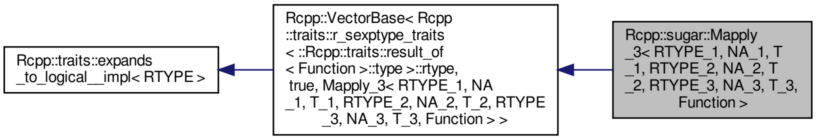
6.376 Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > Class Template Reference

```
#include <mapply_3.h>
```

Inheritance diagram for Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
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::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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
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::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type result_type
- typedef Rcpp::VectorBase< RTYPE_1, NA_1, T_1 > VEC_1
- typedef Rcpp::VectorBase< RTYPE_2, NA_2, T_2 > VEC_2
- typedef Rcpp::VectorBase< RTYPE_3, NA_3, T_3 > VEC_3
- typedef Rcpp::traits::Extractor< RTYPE_1, NA_1, T_1 >::type EXT_1
- typedef Rcpp::traits::Extractor< RTYPE_2, NA_2, T_2 >::type EXT_2
- typedef Rcpp::traits::Extractor< RTYPE_3, NA_3, T_3 >::type EXT_3

Public Member Functions

- Mapply_3 (const VEC_1 &vec_1_, const VEC_2 &vec_2_, const VEC_3 &vec_3_, Function fun_)
- result_type operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT_1 & vec_1
- const EXT_2 & vec_2
- const EXT_3 & vec_3
- Function fun

6.376.1 Detailed Description

template<int RTYPE_1, bool NA_1, typename T_1, int RTYPE_2, bool NA_2, typename T_2, int RTYPE_3, bool NA_3, typename T_3, typename Function>
class Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >

Definition at line 34 of file mapply_3.h.

6.376.2 Member Typedef Documentation

6.376.2.1 EXT_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::traits::Extractor<RTYPE_1,NA_1,T_1>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T↩
_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_1
```

Definition at line 48 of file mapply_3.h.

6.376.2.2 EXT_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::traits::Extractor<RTYPE_2,NA_2,T_2>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T↩
_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_2
```

Definition at line 49 of file mapply_3.h.

6.376.2.3 EXT_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE←
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::traits::Extractor<RTYPE_3,NA_3,T_3>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T←
_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_3
```

Definition at line 50 of file mapply_3.h.

6.376.2.4 result_type

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE←
_3, bool NA_3, typename T_3 , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1,
RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::result_type
```

Definition at line 42 of file mapply_3.h.

6.376.2.5 VEC_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE←
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::VectorBase<RTYPE_1,NA_1,T_1> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2,
NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_1
```

Definition at line 44 of file mapply_3.h.

6.376.2.6 VEC_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE←
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::VectorBase<RTYPE_2,NA_2,T_2> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2,
NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_2
```

Definition at line 45 of file mapply_3.h.

6.376.2.7 VEC_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::VectorBase<RTYPE_3,NA_3,T_3> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2,
NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_3
```

Definition at line 46 of file mapply_3.h.

6.376.3 Constructor & Destructor Documentation

6.376.3.1 Mapply_3()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >↩
::Mapply_3 (
             const VEC_1 & vec_1_,
             const VEC_2 & vec_2_,
             const VEC_3 & vec_3_,
             Function fun_ )  [inline]
```

Definition at line 52 of file mapply_3.h.

6.376.4 Member Function Documentation

6.376.4.1 operator[]()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
result_type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3,
Function >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 55 of file mapply_3.h.

References Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::fun, Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >↩::vec_1, Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >↩::vec_2, and Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::vec_3.

6.376.4.2 size()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function
>::size ( ) const  [inline]
```

Definition at line 58 of file mapply_3.h.

References Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::vec_1.

6.376.5 Member Data Documentation

6.376.5.1 fun

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
Function Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function
>::fun  [private]
```

Definition at line 64 of file mapply_3.h.

Referenced by Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[ ]().

6.376.5.2 vec_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
const EXT_1& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3,
Function >::vec_1  [private]
```

Definition at line 61 of file mapply_3.h.

Referenced by Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[ ](), and Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::size().

6.376.5.3 vec_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
const EXT_2& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3,
Function >::vec_2  [private]
```

Definition at line 62 of file mapply_3.h.

Referenced by Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[ ]().

6.376.5.4 vec_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE↩
_3, bool NA_3, typename T_3 , typename Function >
const EXT_3& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3,
Function >::vec_3  [private]
```

Definition at line 63 of file mapply_3.h.

Referenced by Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[ ]().

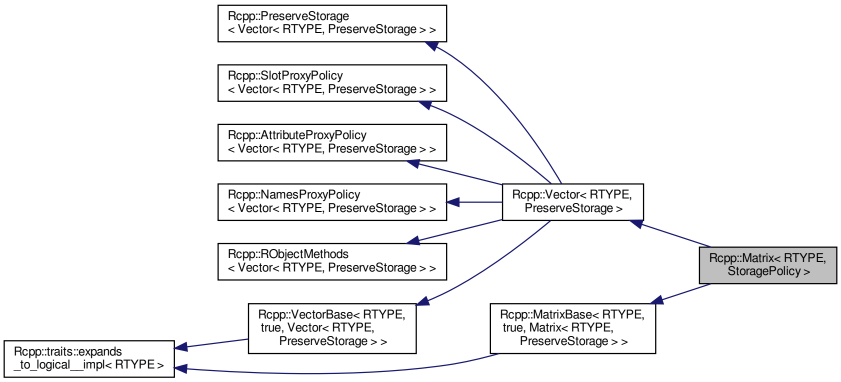
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply_3.h

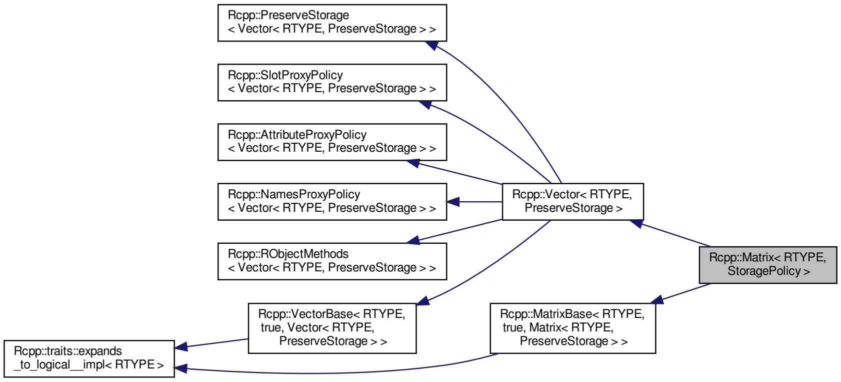
6.377 Rcpp::Matrix< RTYPE, StoragePolicy > Class Template Reference

```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >

Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >

Classes

- struct can_have_na
- struct r_type

Public Types

- typedef MatrixRow< RTYPE > Row
- typedef ConstMatrixRow< RTYPE > ConstRow
- typedef MatrixColumn< RTYPE > Column
- typedef ConstMatrixColumn< RTYPE > ConstColumn
- typedef SubMatrix< RTYPE > Sub
- typedef StoragePolicy< Matrix > Storage
- typedef Vector< RTYPE, StoragePolicy > VECTOR
- typedef VECTOR::iterator iterator
- typedef VECTOR::const_iterator const_iterator
- typedef VECTOR::converter_type converter_type
- typedef VECTOR::stored_type stored_type
- typedef VECTOR::Proxy Proxy
- typedef VECTOR::const_Proxy const_Proxy

Public Member Functions

- Matrix ()
- Matrix (SEXP x)
- Matrix (const Dimension &dims)
- Matrix (const int &nrows_, const int &ncols)
- template<typename Iterator >
  Matrix (const int &nrows_, const int &ncols, Iterator start)
- Matrix (const int &n)

- Matrix (const Matrix &other)
- template<bool NA, typename MAT >
  Matrix (const MatrixBase< RTYPE, NA, MAT > &other)
- Matrix (const SubMatrix< RTYPE > &)
- Matrix & operator= (const Matrix &other)
- Matrix & operator= (const SubMatrix< RTYPE > &)
- Matrix (const no_init_matrix &obj)
- int ncol () const
- int nrow () const
- int cols () const
- int rows () const
- Row row (int i)
- ConstRow row (int i) const
- Column column (int i)
- ConstColumn column (int i) const
- const_iterator begin () const
- const_iterator end () const
- const_iterator cbegin () const
- const_iterator cend () const
- iterator begin ()
- iterator end ()
- template<typename U >
  void fill_diag (const U &u)
- Proxy operator[ ] (R_xlen_t i)
- const_Proxy operator[ ] (R_xlen_t i) const
- Proxy operator() (const size_t &i, const size_t &j)
- const_Proxy operator() (const size_t &i, const size_t &j) const
- Proxy at (const size_t &i, const size_t &j)
- const_Proxy at (const size_t &i, const size_t &j) const
- Row operator() (int i, internal::NamedPlaceHolder)
- ConstRow operator() (int i, internal::NamedPlaceHolder) const
- Column operator() (internal::NamedPlaceHolder, int i)
- ConstColumn operator() (internal::NamedPlaceHolder, int i) const
- Sub operator() (const Range &row_range, const Range &col_range)
- Sub operator() (internal::NamedPlaceHolder, const Range &col_range)
- Sub operator() (const Range &row_range, internal::NamedPlaceHolder)

Static Public Member Functions

- template<typename U >
  static Matrix diag (int size, const U &diag_value)

Private Member Functions

- R_xlen_t offset (const int i, const int j) const
- template<typename U >
  void fill_diag__dispatch (traits::false_type, const U &u)
- template<typename U >
  void fill_diag__dispatch (traits::true_type, const U &u)
- template<bool NA, typename MAT >
  void import_matrix_expression (const MatrixBase< RTYPE, NA, MAT > &other, int nr, int nc)

Private Attributes

- int nrows

Additional Inherited Members

6.377.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::Matrix< RTYPE, StoragePolicy >

Definition at line 28 of file Matrix.h.

6.377.2 Member Typedef Documentation

6.377.2.1 Column

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef MatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Column
```

Definition at line 38 of file Matrix.h.

6.377.2.2 const_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::const_iterator
```

Definition at line 45 of file Matrix.h.

6.377.2.3 const_Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::const_Proxy
```

Definition at line 49 of file Matrix.h.

6.377.2.4 ConstColumn

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ConstMatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstColumn
```

Definition at line 39 of file Matrix.h.

6.377.2.5 ConstRow

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ConstMatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstRow
```

Definition at line 37 of file Matrix.h.

6.377.2.6 converter_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::converter_type Rcpp::Matrix< RTYPE, StoragePolicy >::converter_type
```

Definition at line 46 of file Matrix.h.

6.377.2.7 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::iterator Rcpp::Matrix< RTYPE, StoragePolicy >::iterator
```

Definition at line 44 of file Matrix.h.

6.377.2.8 Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::Proxy
```

Definition at line 48 of file Matrix.h.

6.377.2.9 Row

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef MatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Row
```

Definition at line 36 of file Matrix.h.

6.377.2.10 Storage

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef StoragePolicy<Matrix> Rcpp::Matrix< RTYPE, StoragePolicy >::Storage
```

Definition at line 42 of file Matrix.h.

6.377.2.11 stored_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::stored_type Rcpp::Matrix< RTYPE, StoragePolicy >::stored_type
```

Definition at line 47 of file Matrix.h.

6.377.2.12 Sub

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef SubMatrix<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Sub
```

Definition at line 40 of file Matrix.h.

6.377.2.13 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef Vector<RTYPE, StoragePolicy> Rcpp::Matrix< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 43 of file Matrix.h.

6.377.3 Constructor & Destructor Documentation

6.377.3.1 Matrix() [1/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix ( )  [inline]
```

Definition at line 51 of file Matrix.h.

6.377.3.2 Matrix() [2/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
            SEXP x )  [inline]
```

Definition at line 53 of file Matrix.h.

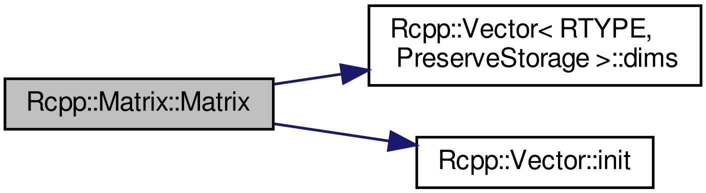
6.377.3.3 Matrix() [3/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
            const Dimension & dims )  [inline]
```

Definition at line 55 of file Matrix.h.

References Rcpp::Vector< RTYPE, PreserveStorage >::dims(), and Rcpp::Vector< RTYPE, StoragePolicy >::init().

Here is the call graph for this function:



6.377.3.4 Matrix() [4/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
            const int & nrows_,
            const int & ncols )  [inline]
```

Definition at line 59 of file Matrix.h.

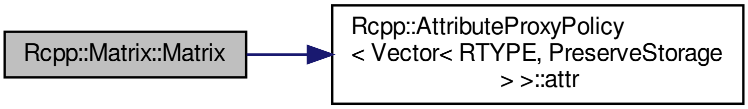
6.377.3.5 Matrix() [5/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename Iterator >
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
            const int & nrows_,
            const int & ncols,
            Iterator start )  [inline]
```

Definition at line 64 of file Matrix.h.

References Rcpp::AttributeProxyPolicy< Vector< RTYPE, PreserveStorage > >::attr(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:


Rcpp::Matrix::Matrix
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >::attr

6.377.3.6 Matrix() [6/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
            const int & n )  [inline]
```

Definition at line 71 of file Matrix.h.

6.377.3.7 Matrix() [7/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
             const Matrix< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 74 of file Matrix.h.

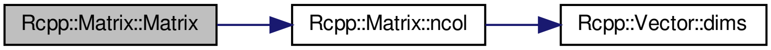
6.377.3.8 Matrix() [8/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename MAT >
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
             const MatrixBase< RTYPE, NA, MAT > & other )  [inline]
```

Definition at line 77 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:



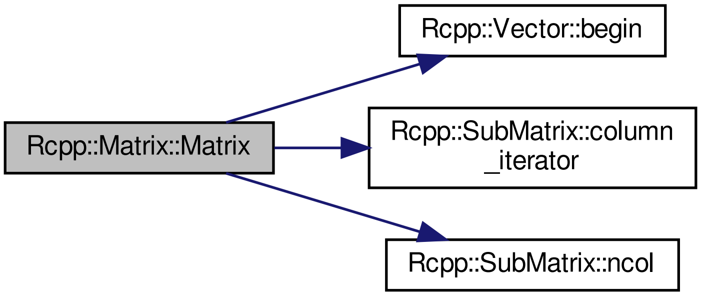
6.377.3.9 Matrix() [9/10]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
             const SubMatrix< RTYPE > & sub )
```

Definition at line 59 of file SubMatrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::SubMatrix< RTYPE >::column_iterator(), Rcpp::SubMatrix< RTYPE >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:



6.377.3.10 Matrix() [10/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
             const no_init_matrix & obj )  [inline], [explicit]
```

Definition at line 92 of file Matrix.h.

6.377.4 Member Function Documentation

6.377.4.1 at() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::at (
             const size_t & i,
             const size_t & j )  [inline]
```

Definition at line 144 of file Matrix.h.

6.377.4.2 at() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::at (
             const size_t & i,
             const size_t & j ) const  [inline]
```

Definition at line 147 of file Matrix.h.

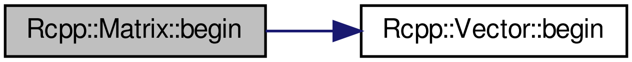
6.377.4.3 begin() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Matrix< RTYPE, StoragePolicy >::begin ( )  [inline]
```

Definition at line 116 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.377.4.4 begin() [2/2]

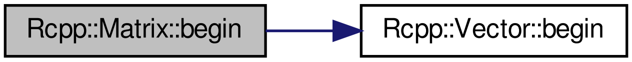
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::begin ( ) const  [inline]
```

Definition at line 112 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Referenced by fastLm().

Here is the call graph for this function:



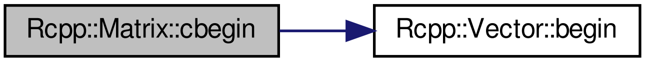
6.377.4.5 cbegin()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::cbegin ( ) const [inline]
```

Definition at line 114 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



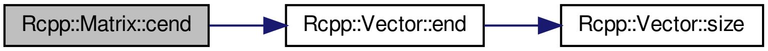
6.377.4.6 cend()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::cend ( ) const [inline]
```

Definition at line 115 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



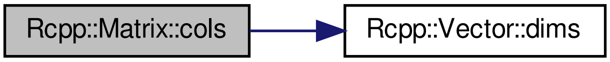
6.377.4.7 cols()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::cols ( ) const  [inline]
```

Definition at line 100 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Here is the call graph for this function:



6.377.4.8 column() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Column Rcpp::Matrix< RTYPE, StoragePolicy >::column (
             int i )  [inline]
```

Definition at line 109 of file Matrix.h.

6.377.4.9 column() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::column (
             int i ) const  [inline]
```

Definition at line 110 of file Matrix.h.

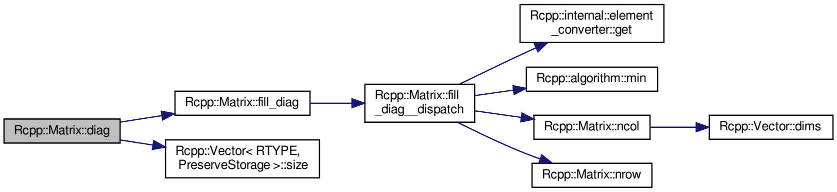
6.377.4.10 diag()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static Matrix Rcpp::Matrix< RTYPE, StoragePolicy >::diag (
             int size,
             const U & diag_value )  [inline], [static]
```

Definition at line 124 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag(), and Rcpp::Vector< RTYPE, PreserveStorage >::size().

Here is the call graph for this function:


Rcpp::Matrix::diag
Rcpp::Matrix::fill_diag
Rcpp::Vector< RTYPE, PreserveStorage >::size
Rcpp::Matrix::fill_diag__dispatch
Rcpp::internal::element_converter::get
Rcpp::algorithm::min
Rcpp::Matrix::ncol
Rcpp::Vector::dims
Rcpp::Matrix::nrow

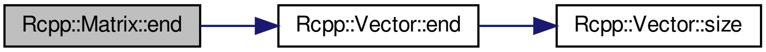
6.377.4.11 end() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Matrix< RTYPE, StoragePolicy >::end ( )  [inline]
```

Definition at line 117 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



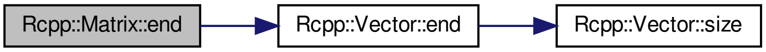
6.377.4.12 end() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::end ( ) const  [inline]
```

Definition at line 113 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



6.377.4.13 fill_diag()

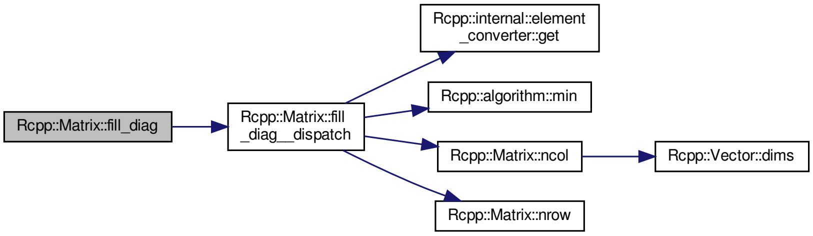
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag (
            const U & u )  [inline]
```

Definition at line 120 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::diag().

Here is the call graph for this function:



6.377.4.14 fill_diag__dispatch() [1/2]

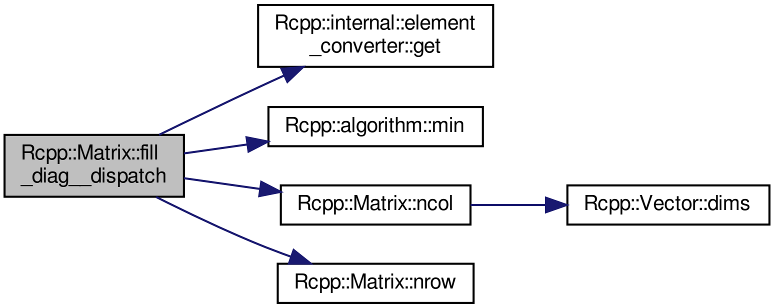
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch (
             traits::false_type ,
             const U & u )  [inline], [private]
```

Definition at line 177 of file Matrix.h.

References Rcpp::internal::element_converter< RTYPE >::get(), Rcpp::algorithm::min(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag().

Here is the call graph for this function:



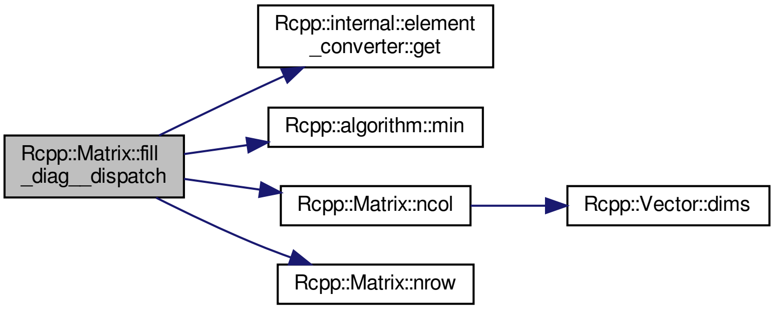
6.377.4.15 fill_diag__dispatch() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch (
             traits::true_type ,
             const U & u )  [inline], [private]
```

Definition at line 187 of file Matrix.h.

References Rcpp::internal::element_converter< RTYPE >::get(), Rcpp::algorithm::min(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:



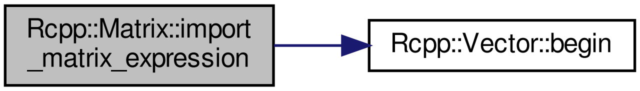
6.377.4.16 import_matrix_expression()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename MAT >
void Rcpp::Matrix< RTYPE, StoragePolicy >::import_matrix_expression (
             const MatrixBase< RTYPE, NA, MAT > & other,
             int nr,
             int nc ) [inline], [private]
```

Definition at line 198 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.377.4.17 ncol()

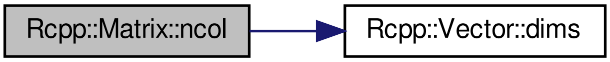
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::ncol ( ) const  [inline]
```

Definition at line 94 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Referenced by Rcpp::internal::as_vector__impl(), fastLm(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator()(), Rcpp::MatrixRow< RTYPE >::size(), Rcpp::ConstMatrixRow< RTYPE >::size(), and Rcpp::trimws().

Here is the call graph for this function:



6.377.4.18 nrow()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::nrow ( ) const  [inline]
```

Definition at line 97 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Referenced by Rcpp::internal::as_vector__impl(), Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow(), fastLm(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag__dispatch(), Rcpp::MatrixRow< RTYPE >::MatrixRow(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator()(), and Rcpp::trimws().

6.377.4.19 offset()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Matrix< RTYPE, StoragePolicy >::offset (
             const int i,
             const int j ) const  [inline], [private]
```

Definition at line 174 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::operator()().

6.377.4.20 operator()() [1/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             const Range & row_range,
             const Range & col_range )  [inline]
```

Definition at line 163 of file Matrix.h.

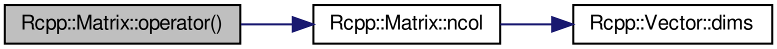
6.377.4.21 operator()() [2/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             const Range & row_range,
             internal::NamedPlaceHolder  )  [inline]
```

Definition at line 169 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:



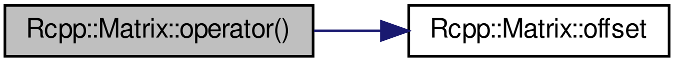
6.377.4.22 operator()() [3/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             const size_t & i,
             const size_t & j )  [inline]
```

Definition at line 137 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



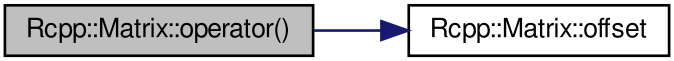
6.377.4.23 operator()() [4/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             const size_t & i,
             const size_t & j ) const  [inline]
```

Definition at line 140 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



6.377.4.24 operator()() [5/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Row Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             int i,
             internal::NamedPlaceHolder  )  [inline]
```

Definition at line 151 of file Matrix.h.

6.377.4.25 operator()() [6/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstRow Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             int i,
             internal::NamedPlaceHolder  ) const  [inline]
```

Definition at line 154 of file Matrix.h.

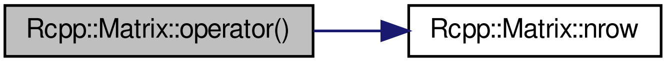
6.377.4.26 operator()() [7/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             internal::NamedPlaceHolder ,
             const Range & col_range )  [inline]
```

Definition at line 166 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:



6.377.4.27 operator()() [8/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Column Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             internal::NamedPlaceHolder ,
             int i )  [inline]
```

Definition at line 157 of file Matrix.h.

6.377.4.28 operator()() [9/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
             internal::NamedPlaceHolder ,
             int i ) const  [inline]
```

Definition at line 160 of file Matrix.h.

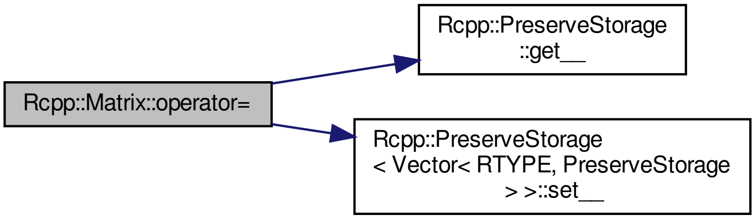
6.377.4.29 operator=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Matrix& Rcpp::Matrix< RTYPE, StoragePolicy >::operator= (
             const Matrix< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 83 of file Matrix.h.

References Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrows, and Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::set__().

Here is the call graph for this function:


Rcpp::Matrix::operator=
Rcpp::PreserveStorage ::get__
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >::set__

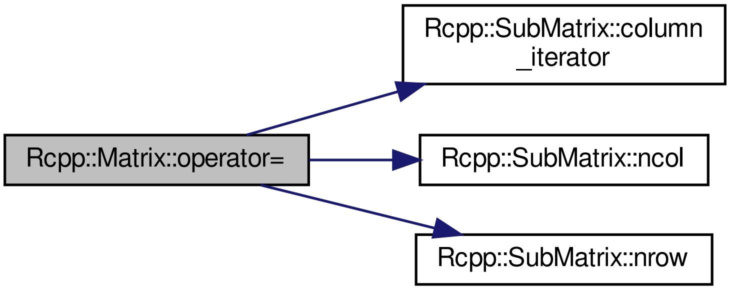
6.377.4.30 operator=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Matrix< RTYPE, StoragePolicy > & Rcpp::Matrix< RTYPE, StoragePolicy >::operator= (
             const SubMatrix< RTYPE > & sub )
```

Definition at line 72 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::column_iterator(), Rcpp::SubMatrix< RTYPE >::ncol(), and Rcpp::SubMatrix< RTYPE >::nrow().

Here is the call graph for this function:



6.377.4.31 operator[]() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[] (
            R_xlen_t i ) [inline]
```

Definition at line 130 of file Matrix.h.

6.377.4.32 operator[]() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 133 of file Matrix.h.

6.377.4.33 row() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Row Rcpp::Matrix< RTYPE, StoragePolicy >::row (
            int i ) [inline]
```

Definition at line 107 of file Matrix.h.

6.377.4.34 row() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstRow Rcpp::Matrix< RTYPE, StoragePolicy >::row (
             int i ) const  [inline]
```

Definition at line 108 of file Matrix.h.

6.377.4.35 rows()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::rows ( ) const  [inline]
```

Definition at line 103 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Referenced by Rcpp::operator<<().

6.377.5 Member Data Documentation

6.377.5.1 nrows

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::nrows  [private]
```

Definition at line 29 of file Matrix.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), Rcpp::Matrix< RTYPE, StoragePolicy >::offset(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::Matrix< RTYPE, StoragePolicy >::rows().

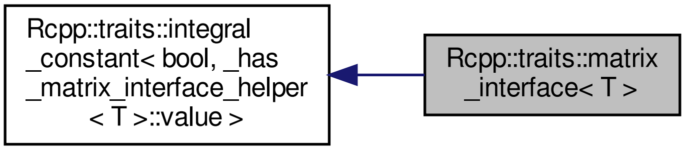
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/Matrix.h
- inst/include/Rcpp/vector/SubMatrix.h

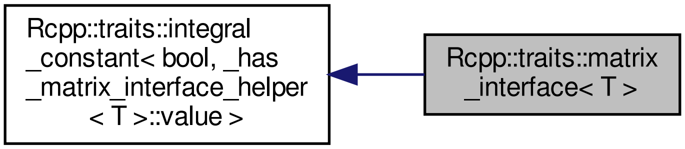
6.378 Rcpp::traits::matrix_interface< T > Struct Template Reference

```
#include <matrix_interface.h>
```

Inheritance diagram for Rcpp::traits::matrix_interface< T >:


Rcpp::traits::integral _constant< bool, _has _matrix_interface_helper < T >::value >
Rcpp::traits::matrix _interface< T >

Collaboration diagram for Rcpp::traits::matrix_interface< T >:


Rcpp::traits::integral _constant< bool, _has _matrix_interface_helper < T >::value >
Rcpp::traits::matrix _interface< T >

Additional Inherited Members

6.378.1 Detailed Description

template<typename T>
struct Rcpp::traits::matrix_interface< T >

Definition at line 46 of file matrix_interface.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/matrix_interface.h

6.379 Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container > Struct Template Reference

```
#include <cbind.h>
```

6.379.1 Detailed Description

template<typename T, bool is_container = has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container >

Definition at line 463 of file cbind.h.

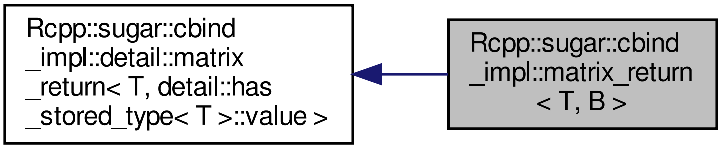
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

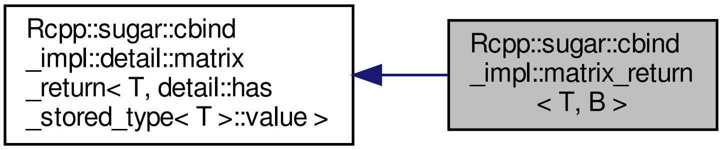
6.380 Rcpp::sugar::cbind_impl::matrix_return< T, B > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::matrix_return< T, B >:


Rcpp::sugar::cbind _impl::detail::matrix _return< T, detail::has _stored_type< T >::value >
Rcpp::sugar::cbind _impl::matrix_return < T, B >

Collaboration diagram for Rcpp::sugar::cbind_impl::matrix_return< T, B >:


Rcpp::sugar::cbind _impl::detail::matrix _return< T, detail::has _stored_type< T >::value >
Rcpp::sugar::cbind _impl::matrix_return < T, B >

6.380.1 Detailed Description

template<typename T, bool B = detail::has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::matrix_return< T, B >

Definition at line 501 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.381 Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false > Struct Reference

```
#include <cbind.h>
```

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.381.1 Detailed Description

Definition at line 488 of file cbind.h.

6.381.2 Member Typedef Documentation

6.381.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false >::type
```

Definition at line 489 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.382 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.382.1 Detailed Description

Definition at line 478 of file cbind.h.

6.382.2 Member Typedef Documentation

6.382.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP
>, true >::type
```

Definition at line 479 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.383 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

Public Types

- typedef Rcpp::Matrix< LGLSXP > type

6.383.1 Detailed Description

Definition at line 483 of file cbind.h.

6.383.2 Member Typedef Documentation

6.383.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP
>, true >::type
```

Definition at line 484 of file cbind.h.

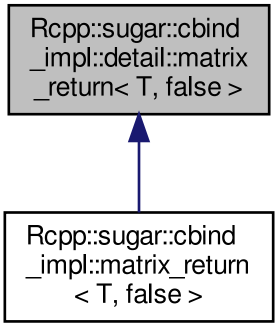
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.384 Rcpp::sugar::cbind_impl::detail::matrix_return< T, false > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >:



Public Types

- enum { RTYPE = cbind_sexptype_traits<T>::rtype }
- typedef Rcpp::Matrix< RTYPE > type

6.384.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >

Definition at line 493 of file cbind.h.

6.384.2 Member Typedef Documentation

6.384.2.1 type

```
template<typename T >
typedef Rcpp::Matrix<RTYPE> Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >::type
```

Definition at line 495 of file cbind.h.

6.384.3 Member Enumeration Documentation

6.384.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	

Definition at line 494 of file cbind.h.

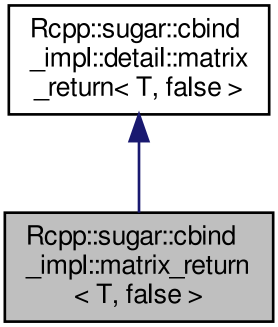
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

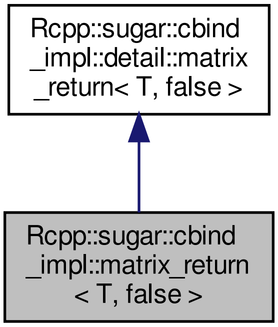
6.385 Rcpp::sugar::cbind_impl::matrix_return< T, false > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::matrix_return< T, false >:



Collaboration diagram for Rcpp::sugar::cbind_impl::matrix_return< T, false >:



Additional Inherited Members

6.385.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::matrix_return< T, false >

Definition at line 505 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.386 Rcpp::sugar::cbind_impl::detail::matrix_return< T, true > Struct Template Reference

```
#include <cbind.h>
```

Public Types

- enum { RTYPE = cbind_sexptype_traits<stored_type>::rtype }
- typedef cbind_storage_type< cbind_sexptype_traits< typename T::stored_type >::rtype >::type stored_type
- typedef Rcpp::Matrix< RTYPE > type

6.386.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >

Definition at line 466 of file cbind.h.

6.386.2 Member Typedef Documentation

6.386.2.1 stored_type

```
template<typename T >
typedef cbind_storage_type< cbind_sexptype_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind_imp
T, true >::stored_type
```

Definition at line 469 of file cbind.h.

6.386.2.2 type

```
template<typename T >
typedef Rcpp::Matrix<RTYPE> Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >::type
```

Definition at line 472 of file cbind.h.

6.386.3 Member Enumeration Documentation

6.386.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	

Definition at line 471 of file cbind.h.

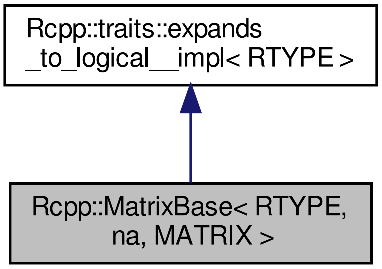
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

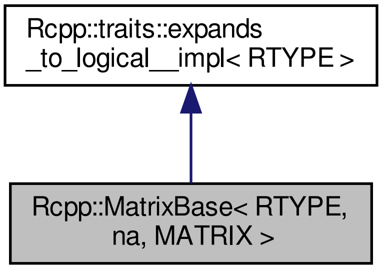
6.387 Rcpp::MatrixBase< RTYPE, na, MATRIX > Class Template Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >:



Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >:



Classes

- struct can_have_na
- class iterator
- struct r_matrix_interface
- struct r_type

Public Types

- typedef traits::storage_type< RTYPE >::type stored_type

Public Member Functions

- MATRIX & get_ref ()
- stored_type operator() (int i, int j) const
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- iterator begin () const
- iterator end () const

Static Public Member Functions

- static MATRIX eye (int n)
- static MATRIX ones (int n)
- static MATRIX zeros (int n)

6.387.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
class Rcpp::MatrixBase< RTYPE, na, MATRIX >

a base class for vectors, modelled after the CRTP

Definition at line 32 of file MatrixBase.h.

6.387.2 Member Typedef Documentation

6.387.2.1 stored_type

```
template<int RTYPE, bool na, typename MATRIX >
typedef traits::storage_type<RTYPE>::type Rcpp::MatrixBase< RTYPE, na, MATRIX >::stored_type
```

Definition at line 37 of file MatrixBase.h.

6.387.3 Member Function Documentation

6.387.3.1 begin()

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::begin ( ) const  [inline]
```

Definition at line 191 of file MatrixBase.h.

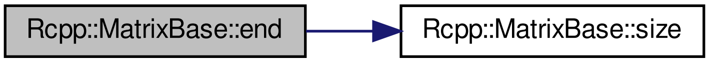
6.387.3.2 end()

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::end ( ) const  [inline]
```

Definition at line 192 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::size().

Here is the call graph for this function:



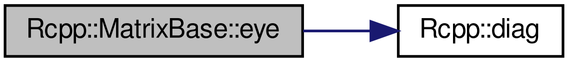
6.387.3.3 eye()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::eye (
             int n )  [inline], [static]
```

Definition at line 51 of file MatrixBase.h.

References Rcpp::diag().

Here is the call graph for this function:



6.387.3.4 get_ref()

```
template<int RTYPE, bool na, typename MATRIX >
MATRIX& Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref ( )  [inline]
```

Definition at line 39 of file MatrixBase.h.

Referenced by Rcpp::internal::as_vector__impl().

6.387.3.5 ncol()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol ( ) const  [inline]
```

Definition at line 49 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::Row↩ MeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp↩ ::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

6.387.3.6 nrow()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow ( ) const  [inline]
```

Definition at line 48 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

6.387.3.7 ones()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::ones (
             int n )  [inline], [static]
```

Definition at line 60 of file MatrixBase.h.

6.387.3.8 operator()()

```
template<int RTYPE, bool na, typename MATRIX >
stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 43 of file MatrixBase.h.

6.387.3.9 size()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::size ( ) const  [inline]
```

Definition at line 47 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::end().

6.387.3.10 zeros()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::zeros (
             int n )  [inline], [static]
```

Definition at line 74 of file MatrixBase.h.

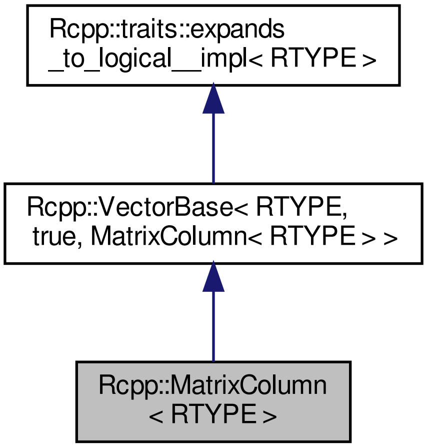
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

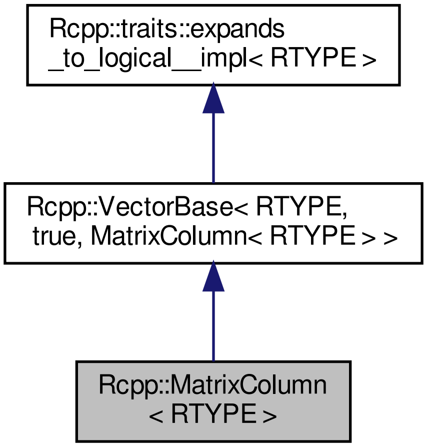
6.388 Rcpp::MatrixColumn< RTYPE > Class Template Reference

```
#include <MatrixColumn.h>
```

Inheritance diagram for Rcpp::MatrixColumn< RTYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, MatrixColumn< RTYPE > >
Rcpp::MatrixColumn < RTYPE >

Collaboration diagram for Rcpp::MatrixColumn< RTYPE >:



Public Types

- typedef Matrix< RTYPE > MATRIX
- typedef MATRIX::Proxy Proxy
- typedef MATRIX::const_Proxy const_Proxy
- typedef MATRIX::value_type value_type
- typedef MATRIX::iterator iterator
- typedef MATRIX::const_iterator const_iterator

Public Member Functions

- MatrixColumn (MATRIX &parent, int i)
- MatrixColumn (const MATRIX &parent, int i)
- MatrixColumn (const MatrixColumn &other)
- template< int RT, bool NA, typename T >
  MatrixColumn & operator= (const Rcpp::VectorBase< RT, NA, T > &rhs)
- MatrixColumn & operator= (const MatrixColumn &rhs)
- Proxy operator[ ] (int i)
- const_Proxy operator[ ] (int i) const
- const_iterator begin () const
- const_iterator end () const
- const_iterator cbegin () const
- const_iterator cend () const
- iterator begin ()
- iterator end ()
- int size () const

Private Attributes

- const int n
- iterator start
- const_iterator const_start

6.388.1 Detailed Description

template<int RTYPE>
class Rcpp::MatrixColumn< RTYPE >

Definition at line 28 of file MatrixColumn.h.

6.388.2 Member Typedef Documentation

6.388.2.1 const_iterator

```
template<int RTYPE>
typedef MATRIX::const_iterator Rcpp::MatrixColumn< RTYPE >::const_iterator
```

Definition at line 35 of file MatrixColumn.h.

6.388.2.2 const_Proxy

```
template<int RTYPE>
typedef MATRIX::const_Proxy Rcpp::MatrixColumn< RTYPE >::const_Proxy
```

Definition at line 32 of file MatrixColumn.h.

6.388.2.3 iterator

```
template<int RTYPE>
typedef MATRIX::iterator Rcpp::MatrixColumn< RTYPE >::iterator
```

Definition at line 34 of file MatrixColumn.h.

6.388.2.4 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::MatrixColumn< RTYPE >::MATRIX
```

Definition at line 30 of file MatrixColumn.h.

6.388.2.5 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixColumn< RTYPE >::Proxy
```

Definition at line 31 of file MatrixColumn.h.

6.388.2.6 value_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::MatrixColumn< RTYPE >::value_type
```

Definition at line 33 of file MatrixColumn.h.

6.388.3 Constructor & Destructor Documentation

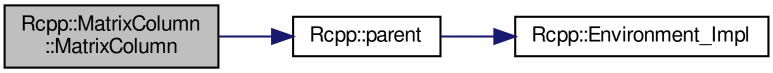
6.388.3.1 MatrixColumn() [1/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
             MATRIX & parent,
             int i )  [inline]
```

Definition at line 37 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:



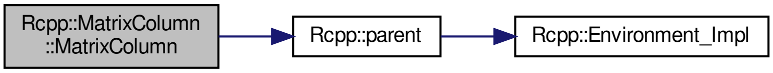
6.388.3.2 MatrixColumn() [2/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
             const MATRIX & parent,
             int i )  [inline]
```

Definition at line 49 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:



6.388.3.3 MatrixColumn() [3/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
             const MatrixColumn< RTYPE > & other )  [inline]
```

Definition at line 61 of file MatrixColumn.h.

6.388.4 Member Function Documentation

6.388.4.1 begin() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::begin ( )  [inline]
```

Definition at line 103 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

6.388.4.2 begin() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::begin ( ) const  [inline]
```

Definition at line 87 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.388.4.3 cbegin()

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::cbegin ( ) const  [inline]
```

Definition at line 95 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.388.4.4 cend()

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::cend ( ) const  [inline]
```

Definition at line 99 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start, and Rcpp::MatrixColumn< RTYPE >::n.

6.388.4.5 end() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::end ( )  [inline]
```

Definition at line 107 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n, and Rcpp::MatrixColumn< RTYPE >::start.

6.388.4.6 end() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::end ( ) const  [inline]
```

Definition at line 91 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start, and Rcpp::MatrixColumn< RTYPE >::n.

6.388.4.7 operator=() [1/2]

```
template<int RTYPE>
MatrixColumn& Rcpp::MatrixColumn< RTYPE >::operator= (
             const MatrixColumn< RTYPE > & rhs )  [inline]
```

Definition at line 73 of file MatrixColumn.h.

References RCPP_LOOP_UNROLL, and Rcpp::MatrixColumn< RTYPE >::start.

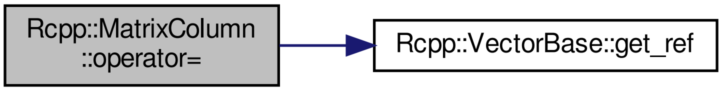
6.388.4.8 operator=() [2/2]

```
template<int RTYPE>
template<int RT, bool NA, typename T >
MatrixColumn& Rcpp::MatrixColumn< RTYPE >::operator= (
             const Rcpp::VectorBase< RT, NA, T > & rhs )  [inline]
```

Definition at line 67 of file MatrixColumn.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), RCPP_LOOP_UNROLL, and Rcpp::MatrixColumn< RTYPE >::start.

Here is the call graph for this function:



6.388.4.9 operator[]() [1/2]

```
template<int RTYPE>
Proxy Rcpp::MatrixColumn< RTYPE >::operator[] (
            int i ) [inline]
```

Definition at line 79 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

6.388.4.10 operator[]() [2/2]

```
template<int RTYPE>
const_Proxy Rcpp::MatrixColumn< RTYPE >::operator[] (
            int i ) const [inline]
```

Definition at line 83 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const_start.

6.388.4.11 size()

```
template<int RTYPE>
int Rcpp::MatrixColumn< RTYPE >::size ( ) const [inline]
```

Definition at line 111 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n.

6.388.5 Member Data Documentation

6.388.5.1 const_start

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::const_start [private]
```

Definition at line 118 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::MatrixColumn< RTYPE >::cbegin(), Rcpp::MatrixColumn< RTYPE >::cend(), Rcpp::MatrixColumn< RTYPE >::end(), and Rcpp::MatrixColumn< RTYPE >::operator[ ]().

6.388.5.2 n

```
template<int RTYPE>
const int Rcpp::MatrixColumn< RTYPE >::n  [private]
```

Definition at line 116 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::cend(), Rcpp::MatrixColumn< RTYPE >::end(), and Rcpp::MatrixColumn< RTYPE >::size().

6.388.5.3 start

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::start  [private]
```

Definition at line 117 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::MatrixColumn< RTYPE >::end(), Rcpp::MatrixColumn< RTYPE >::operator=(), and Rcpp::MatrixColumn< RTYPE >::operator[ ]().

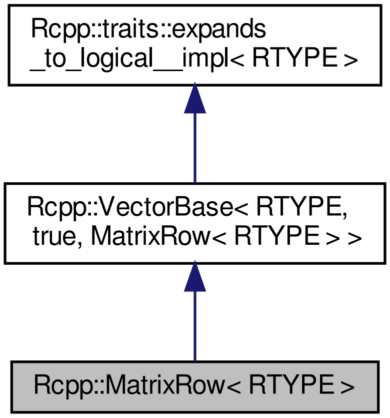
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

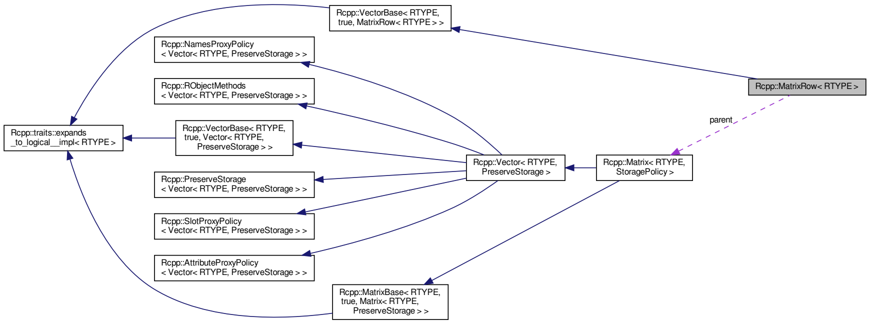
6.389 Rcpp::MatrixRow< RTYPE > Class Template Reference

```
#include <MatrixRow.h>
```

Inheritance diagram for Rcpp::MatrixRow< RTYPE >:



Collaboration diagram for Rcpp::MatrixRow< RTYPE >:


Rcpp::VectorBase< RTYPE, true, MatrixRow< RTYPE > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
Rcpp::MatrixRow< RTYPE >
parent

Classes

- struct const_iter_traits
- class iter_base
- struct iter_traits

Public Types

- typedef Matrix< RTYPE > MATRIX
- typedef MATRIX::Proxy Proxy
- typedef MATRIX::Proxy reference
- typedef MATRIX::const_Proxy const_reference
- typedef MATRIX::value_type value_type
- typedef iter_base< iter_traits > iterator
- typedef iter_base< const_iter_traits > const_iterator

Public Member Functions

- MatrixRow (MATRIX &object, int i)
- MatrixRow (const MatrixRow &other)
- template<int RT, bool NA, typename T >
  MatrixRow & operator= (const Rcpp::VectorBase< RT, NA, T > &rhs)
- MatrixRow & operator= (const MatrixRow &rhs)
- reference operator[ ] (int i)
- reference operator[ ] (int i) const
- iterator begin ()
- iterator end ()
- const_iterator begin () const
- const_iterator end () const
- const_iterator cbegin () const
- const_iterator cend () const
- int size () const

Private Member Functions

- int get_parent_index (int i) const

Private Attributes

- MATRIX & parent
- MATRIX::iterator start
- int parent_nrow
- int row

6.389.1 Detailed Description

template<int RTYPE>
class Rcpp::MatrixRow< RTYPE >

Definition at line 28 of file MatrixRow.h.

6.389.2 Member Typedef Documentation

6.389.2.1 const_iterator

```
template<int RTYPE>
typedef iter_base< const_iter_traits > Rcpp::MatrixRow< RTYPE >::const_iterator
```

Definition at line 131 of file MatrixRow.h.

6.389.2.2 const_reference

```
template<int RTYPE>
typedef MATRIX::const_Proxy Rcpp::MatrixRow< RTYPE >::const_reference
```

Definition at line 33 of file MatrixRow.h.

6.389.2.3 iterator

```
template<int RTYPE>
typedef iter_base< iter_traits > Rcpp::MatrixRow< RTYPE >::iterator
```

Definition at line 130 of file MatrixRow.h.

6.389.2.4 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::MatrixRow< RTYPE >::MATRIX
```

Definition at line 30 of file MatrixRow.h.

6.389.2.5 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::Proxy
```

Definition at line 31 of file MatrixRow.h.

6.389.2.6 reference

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::reference
```

Definition at line 32 of file MatrixRow.h.

6.389.2.7 value_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::MatrixRow< RTYPE >::value_type
```

Definition at line 34 of file MatrixRow.h.

6.389.3 Constructor & Destructor Documentation

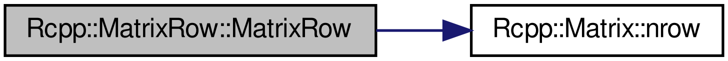
6.389.3.1 MatrixRow() [1/2]

```
template<int RTYPE>
Rcpp::MatrixRow< RTYPE >::MatrixRow (
            MATRIX & object,
            int i )  [inline]
```

Definition at line 133 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::MatrixRow< RTYPE >::parent.

Here is the call graph for this function:



6.389.3.2 MatrixRow() [2/2]

```
template<int RTYPE>
Rcpp::MatrixRow< RTYPE >::MatrixRow (
            const MatrixRow< RTYPE > & other )  [inline]
```

Definition at line 146 of file MatrixRow.h.

6.389.4 Member Function Documentation

6.389.4.1 begin() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixRow< RTYPE >::begin ( )  [inline]
```

Definition at line 175 of file MatrixRow.h.

6.389.4.2 begin() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::begin ( ) const  [inline]
```

Definition at line 183 of file MatrixRow.h.

6.389.4.3 cbegin()

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::cbegin ( ) const  [inline]
```

Definition at line 191 of file MatrixRow.h.

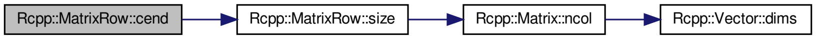
6.389.4.4 cend()

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::cend ( ) const  [inline]
```

Definition at line 195 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



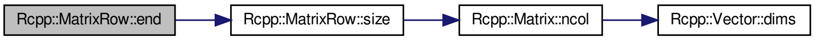
6.389.4.5 end() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixRow< RTYPE >::end ( )  [inline]
```

Definition at line 179 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



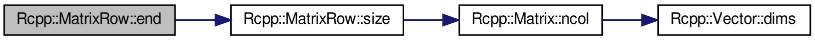
6.389.4.6 end() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::end ( ) const  [inline]
```

Definition at line 187 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



6.389.4.7 get_parent_index()

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::get_parent_index (
             int i ) const  [inline], [private]
```

Definition at line 209 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent_nrow, and RCPP_DEBUG_4.

Referenced by Rcpp::MatrixRow< RTYPE >::operator=(), and Rcpp::MatrixRow< RTYPE >::operator[ ]().

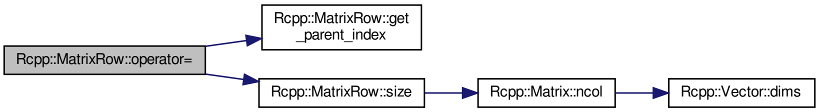
6.389.4.8 operator=() [1/2]

```
template<int RTYPE>
MatrixRow& Rcpp::MatrixRow< RTYPE >::operator= (
             const MatrixRow< RTYPE > & rhs )  [inline]
```

Definition at line 161 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get_parent_index(), RCPP_LOOP_UNROLL_LHSFUN, Rcpp::MatrixRow< RTYPE >::size(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



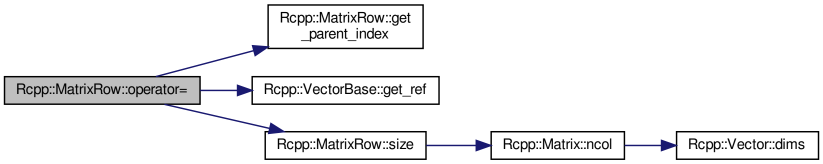
6.389.4.9 operator=() [2/2]

```
template<int RTYPE>
template<int RT, bool NA, typename T >
MatrixRow& Rcpp::MatrixRow< RTYPE >::operator= (
             const Rcpp::VectorBase< RT, NA, T > & rhs )  [inline]
```

Definition at line 154 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get_parent_index(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), RCPP_LOOP_UNROLL_LHSFUN, Rcpp::MatrixRow< RTYPE >::size(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



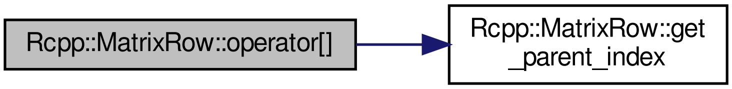
6.389.4.10 operator[]() [1/2]

```
template<int RTYPE>
reference Rcpp::MatrixRow< RTYPE >::operator[] (
             int i )  [inline]
```

Definition at line 167 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get_parent_index(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



6.389.4.11 operator[]() [2/2]

```
template<int RTYPE>
reference Rcpp::MatrixRow< RTYPE >::operator[] (
             int i ) const  [inline]
```

Definition at line 171 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent, Rcpp::MatrixRow< RTYPE >::parent_nrow, and Rcpp::MatrixRow< RTYPE >::row.

6.389.4.12 size()

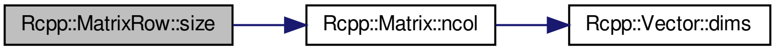
```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::size ( ) const  [inline]
```

Definition at line 199 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::MatrixRow< RTYPE >::parent.

Referenced by Rcpp::MatrixRow< RTYPE >::cend(), Rcpp::MatrixRow< RTYPE >::end(), and Rcpp::MatrixRow< RTYPE >::operator=().

Here is the call graph for this function:



6.389.5 Member Data Documentation

6.389.5.1 parent

```
template<int RTYPE>
MATRIX& Rcpp::MatrixRow< RTYPE >::parent  [private]
```

Definition at line 204 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::MatrixRow(), Rcpp::MatrixRow< RTYPE >::operator[](), and Rcpp::MatrixRow< RTYPE >::size().

6.389.5.2 parent_nrow

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::parent_nrow [private]
```

Definition at line 206 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::get_parent_index(), and Rcpp::MatrixRow< RTYPE >::operator[ ]().

6.389.5.3 row

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::row [private]
```

Definition at line 207 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::operator[ ]().

6.389.5.4 start

```
template<int RTYPE>
MATRIX::iterator Rcpp::MatrixRow< RTYPE >::start [private]
```

Definition at line 205 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::operator=(), and Rcpp::MatrixRow< RTYPE >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

6.390 Rcpp::sugar::Max< RTYPE, NA, T > Class Template Reference

```
#include <max.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Max (const T &obj_)
- operator STORAGE () const

Private Attributes

- const T & obj

6.390.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Max< RTYPE, NA, T >

Definition at line 28 of file max.h.

6.390.2 Member Typedef Documentation

6.390.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Max< RTYPE, NA, T >::STORAGE
```

Definition at line 30 of file max.h.

6.390.3 Constructor & Destructor Documentation

6.390.3.1 Max()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Max< RTYPE, NA, T >::Max (
             const T & obj_ )  [inline]
```

Definition at line 32 of file max.h.

6.390.4 Member Function Documentation

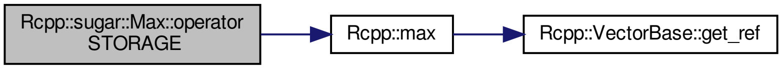
6.390.4.1 operator STORAGE()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE ( ) const  [inline]
```

Definition at line 34 of file max.h.

References Rcpp::max(), and Rcpp::sugar::Max< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.390.5 Member Data Documentation

6.390.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Max< RTYPE, NA, T >::obj  [private]
```

Definition at line 50 of file max.h.

Referenced by Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE(), and Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/max.h

6.391 Rcpp::sugar::Max< RTYPE, false, T > Class Template Reference

```
#include <max.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Max (const T &obj_)
- operator STORAGE () const

Private Attributes

- const T & obj

6.391.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Max< RTYPE, false, T >

Definition at line 55 of file max.h.

6.391.2 Member Typedef Documentation

6.391.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Max< RTYPE, false, T >::STORAGE
```

Definition at line 57 of file max.h.

6.391.3 Constructor & Destructor Documentation

6.391.3.1 Max()

```
template<int RTYPE, typename T >
Rcpp::sugar::Max< RTYPE, false, T >::Max (
            const T & obj_ )  [inline]
```

Definition at line 59 of file max.h.

6.391.4 Member Function Documentation

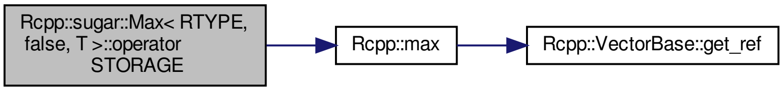
6.391.4.1 operator STORAGE()

```
template<int RTYPE, typename T >
Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE ( ) const  [inline]
```

Definition at line 61 of file max.h.

References Rcpp::max(), and Rcpp::sugar::Max< RTYPE, NA, T >::obj.

Here is the call graph for this function:


Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE
Rcpp::max
Rcpp::VectorBase::get_ref

6.391.5 Member Data Documentation

6.391.5.1 obj

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Max< RTYPE, false, T >::obj  [private]
```

Definition at line 75 of file max.h.

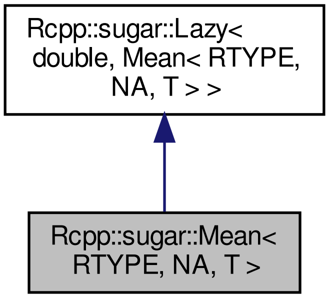
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/max.h

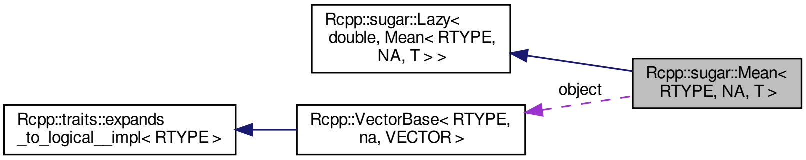
6.392 Rcpp::sugar::Mean< RTYPE, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< RTYPE, NA, T >:


Rcpp::sugar::Lazy< double, Mean< RTYPE, NA, T > >
Rcpp::sugar::Mean< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Mean< RTYPE, NA, T >:


Rcpp::sugar::Lazy< double, Mean< RTYPE, NA, T > >
Rcpp::sugar::Mean< RTYPE, NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::Vector< RTYPE > VECTOR

Public Member Functions

- Mean (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_TYPE & object

6.392.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Mean< RTYPE, NA, T >

Definition at line 29 of file mean.h.

6.392.2 Member Typedef Documentation

6.392.2.1 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Mean< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file mean.h.

6.392.2.2 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Mean< RTYPE, NA, T >::VECTOR
```

Definition at line 32 of file mean.h.

6.392.3 Constructor & Destructor Documentation

6.392.3.1 Mean()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Mean< RTYPE, NA, T >::Mean (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 34 of file mean.h.

6.392.4 Member Function Documentation

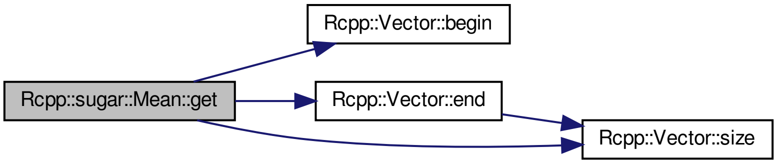
6.392.4.1 get()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::sugar::Mean< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 36 of file mean.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::sugar::Mean< RTYPE, NA, T >::object, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.392.5 Member Data Documentation

6.392.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< RTYPE, NA, T >::object  [private]
```

Definition at line 51 of file mean.h.

Referenced by Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< CPLXSXP, NA, T >::get(), Rcpp::sugar::Mean< LGLSXP, NA, T >::get(), and Rcpp::sugar::Mean< INTSXP, NA, T >::get().

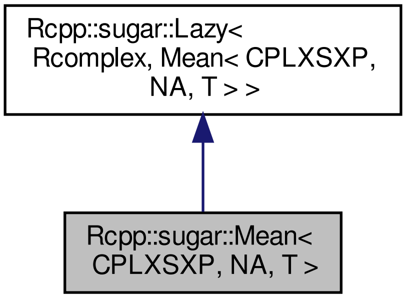
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

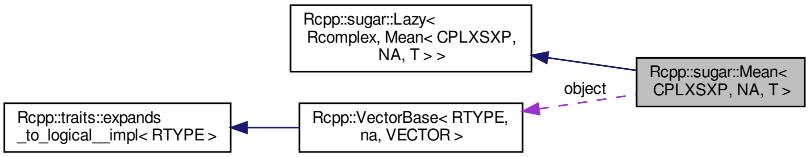
6.393 Rcpp::sugar::Mean< CPLXSXP, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< CPLXSXP, NA, T >:


Rcpp::sugar::Lazy< Rcomplex, Mean< CPLXSXP, NA, T > >
Rcpp::sugar::Mean< CPLXSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Mean< CPLXSXP, NA, T >:


Rcpp::sugar::Lazy< Rcomplex, Mean< CPLXSXP, NA, T > >
Rcpp::sugar::Mean< CPLXSXP, NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

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

6.393.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< CPLXSXP, NA, T >

Definition at line 55 of file mean.h.

6.393.2 Member Typedef Documentation

6.393.2.1 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::Mean< CPLXSXP, NA, T >::VEC_TYPE
```

Definition at line 57 of file mean.h.

6.393.3 Constructor & Destructor Documentation

6.393.3.1 Mean()

```
template<bool NA, typename T >
Rcpp::sugar::Mean< CPLXSXP, NA, T >::Mean (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 59 of file mean.h.

6.393.4 Member Function Documentation

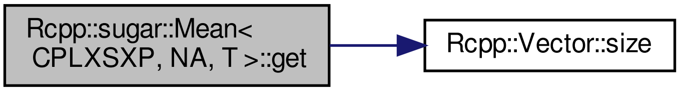
6.393.4.1 get()

```
template<bool NA, typename T >
Rcomplex Rcpp::sugar::Mean< CPLXSXP, NA, T >::get ( ) const  [inline]
```

Definition at line 61 of file mean.h.

References Rcpp::sugar::Mean< RTYPE, NA, T >::object, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.393.5 Member Data Documentation

6.393.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< CPLXSXP, NA, T >::object  [private]
```

Definition at line 88 of file mean.h.

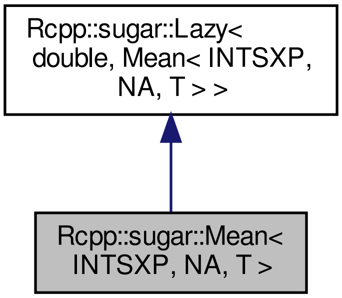
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

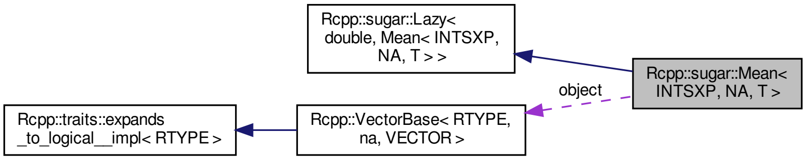
6.394 Rcpp::sugar::Mean< INTSXP, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< INTSXP, NA, T >:


Rcpp::sugar::Lazy< double, Mean< INTSXP, NA, T > >
Rcpp::sugar::Mean< INTSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Mean< INTSXP, NA, T >:


Rcpp::sugar::Lazy< double, Mean< INTSXP, NA, T > >
Rcpp::sugar::Mean< INTSXP, NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< INTSXP, NA, T > VEC_TYPE

Public Member Functions

- Mean (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_TYPE & object

6.394.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< INTSXP, NA, T >

Definition at line 114 of file mean.h.

6.394.2 Member Typedef Documentation

6.394.2.1 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<INTSXP,NA,T> Rcpp::sugar::Mean< INTSXP, NA, T >::VEC_TYPE
```

Definition at line 116 of file mean.h.

6.394.3 Constructor & Destructor Documentation

6.394.3.1 Mean()

```
template<bool NA, typename T >
Rcpp::sugar::Mean< INTSXP, NA, T >::Mean (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 118 of file mean.h.

6.394.4 Member Function Documentation

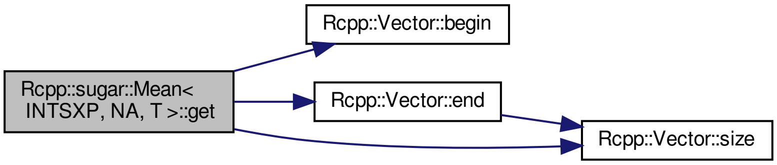
6.394.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Mean< INTSXP, NA, T >::get ( ) const  [inline]
```

Definition at line 120 of file mean.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp↩ ::sugar::Mean< RTYPE, NA, T >::object, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.394.5 Member Data Documentation

6.394.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< INTSXP, NA, T >::object  [private]
```

Definition at line 134 of file mean.h.

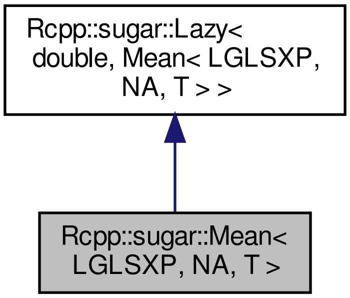
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

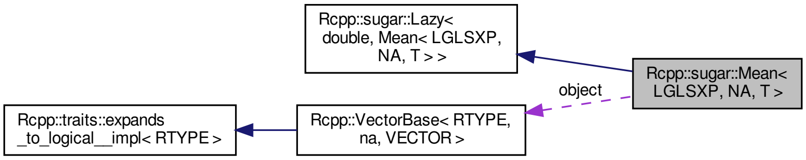
6.395 Rcpp::sugar::Mean< LGLSXP, NA, T > Class Template Reference

`#include <mean.h>`

Inheritance diagram for Rcpp::sugar::Mean< LGLSXP, NA, T >:


Rcpp::sugar::Lazy< double, Mean< LGLSXP, NA, T > >
Rcpp::sugar::Mean< LGLSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Mean< LGLSXP, NA, T >:


Rcpp::sugar::Lazy< double, Mean< LGLSXP, NA, T > >
Rcpp::sugar::Mean< LGLSXP, NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, NA, T > VEC_TYPE

Public Member Functions

- Mean (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_TYPE & object

6.395.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Mean< LGLSXP, NA, T >

Definition at line 92 of file mean.h.

6.395.2 Member Typedef Documentation

6.395.2.1 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::Mean< LGLSXP, NA, T >::VEC_TYPE
```

Definition at line 94 of file mean.h.

6.395.3 Constructor & Destructor Documentation

6.395.3.1 Mean()

```
template<bool NA, typename T >
Rcpp::sugar::Mean< LGLSXP, NA, T >::Mean (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 96 of file mean.h.

6.395.4 Member Function Documentation

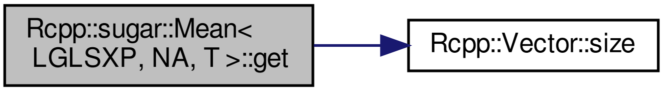
6.395.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Mean< LGLSXP, NA, T >::get ( ) const  [inline]
```

Definition at line 98 of file mean.h.

References Rcpp::sugar::Mean< RTYPE, NA, T >::object, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.395.5 Member Data Documentation

6.395.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< LGLSXP, NA, T >::object  [private]
```

Definition at line 110 of file mean.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mean.h

6.396 Rcpp::sugar::Median< RTYPE, NA, T, NA_RM > Class Template Reference

```
#include <median.h>
```

Public Types

- enum { RESULT_RTYPE = median_detail::result<RTYPE>::rtype }
- typedef median_detail::result< RTYPE >::type result_type
- typedef Rcpp::traits::storage_type< RTYPE >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.396.1 Detailed Description

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >

Definition at line 87 of file median.h.

6.396.2 Member Typedef Documentation

6.396.2.1 result_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef median_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::result_type
```

Definition at line 89 of file median.h.

6.396.2.2 stored_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >↩
::stored_type
```

Definition at line 90 of file median.h.

6.396.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef T Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::VECTOR
```

Definition at line 92 of file median.h.

6.396.3 Member Enumeration Documentation

6.396.3.1 anonymous enum

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
anonymous enum
```

Enumerator

RESULT_RTYPE	

Definition at line 91 of file median.h.

6.396.4 Constructor & Destructor Documentation

6.396.4.1 Median()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::Median (
             const VECTOR & xx )  [inline]
```

Definition at line 98 of file median.h.

6.396.5 Member Function Documentation

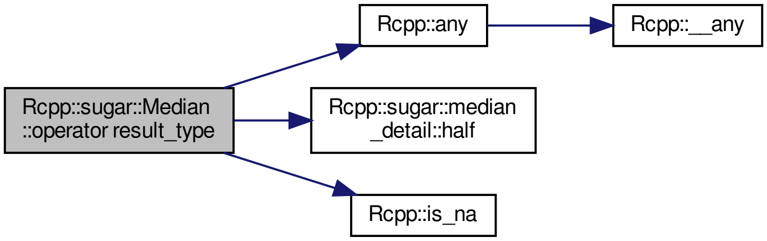
6.396.5.1 operator result_type()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::operator result_type ( )  [inline]
```

Definition at line 101 of file median.h.

References Rcpp::any(), Rcpp::sugar::median_detail::half(), Rcpp::is_na(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.396.6 Member Data Documentation

6.396.6.1 x

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
VECTOR Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x  [private]
```

Definition at line 95 of file median.h.

Referenced by Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::operator result_type(), Rcpp::sugar::Median< RTYPE, NA, T, true >::operator result_type(), Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::operator result_type(), Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type(), Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result_type(), and Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.397 Rcpp::sugar::Median< RTYPE, false, T, NA_RM > Class Template Reference

```
#include <median.h>
```

Public Types

- enum { RESULT_RTYPE = median_detail::result<RTYPE>::rtype }
- typedef median_detail::result< RTYPE >::type result_type
- typedef Rcpp::traits::storage_type< RTYPE >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.397.1 Detailed Description

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::Median< RTYPE, false, T, NA_RM >

Definition at line 159 of file median.h.

6.397.2 Member Typedef Documentation

6.397.2.1 result_type

```
template<int RTYPE, typename T , bool NA_RM>
typedef median_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::result_type
```

Definition at line 161 of file median.h.

6.397.2.2 stored_type

```
template<int RTYPE, typename T , bool NA_RM>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, false, T, NA_RM >↩
::stored_type
```

Definition at line 162 of file median.h.

6.397.2.3 VECTOR

```
template<int RTYPE, typename T , bool NA_RM>
typedef T Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::VECTOR
```

Definition at line 164 of file median.h.

6.397.3 Member Enumeration Documentation

6.397.3.1 anonymous enum

```
template<int RTYPE, typename T , bool NA_RM>
anonymous enum
```

Enumerator

RESULT_RTYPE	

Definition at line 163 of file median.h.

6.397.4 Constructor & Destructor Documentation

6.397.4.1 Median()

```
template<int RTYPE, typename T , bool NA_RM>
Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::Median (
             const VECTOR & xx )  [inline]
```

Definition at line 170 of file median.h.

6.397.5 Member Function Documentation

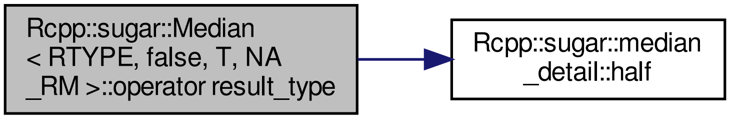
6.397.5.1 operator result_type()

```
template<int RTYPE, typename T , bool NA_RM>
Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::operator result_type ( )  [inline]
```

Definition at line 173 of file median.h.

References Rcpp::sugar::median_detail::half(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.397.6 Member Data Documentation

6.397.6.1 x

```
template<int RTYPE, typename T , bool NA_RM>
VECTOR Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::x  [private]
```

Definition at line 167 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.398 Rcpp::sugar::Median< RTYPE, NA, T, true > Class Template Reference

```
#include <median.h>
```

Public Types

- enum { RESULT_RTYPE = median_detail::result<RTYPE>::rtype }
- typedef median_detail::result< RTYPE >::type result_type
- typedef Rcpp::traits::storage_type< RTYPE >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.398.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Median< RTYPE, NA, T, true >

Definition at line 125 of file median.h.

6.398.2 Member Typedef Documentation

6.398.2.1 result_type

```
template<int RTYPE, bool NA, typename T >
typedef median_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, true >::result_type
```

Definition at line 127 of file median.h.

6.398.2.2 stored_type

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, true >↩
::stored_type
```

Definition at line 128 of file median.h.

6.398.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >
typedef T Rcpp::sugar::Median< RTYPE, NA, T, true >::VECTOR
```

Definition at line 130 of file median.h.

6.398.3 Member Enumeration Documentation

6.398.3.1 anonymous enum

```
template<int RTYPE, bool NA, typename T >
anonymous enum
```

Enumerator

RESULT_RTYPE	

Definition at line 129 of file median.h.

6.398.4 Constructor & Destructor Documentation

6.398.4.1 Median()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Median< RTYPE, NA, T, true >::Median (
            const VECTOR & xx )  [inline]
```

Definition at line 136 of file median.h.

6.398.5 Member Function Documentation

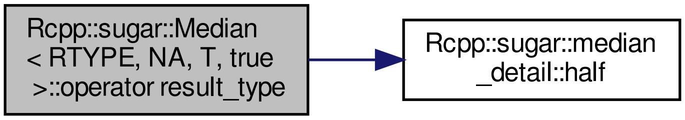
6.398.5.1 operator result_type()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Median< RTYPE, NA, T, true >::operator result_type ( )  [inline]
```

Definition at line 139 of file median.h.

References Rcpp::sugar::median_detail::half(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.398.6 Member Data Documentation

6.398.6.1 x

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Median< RTYPE, NA, T, true >::x  [private]
```

Definition at line 133 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.399 Rcpp::sugar::Median< STRSXP, false, T, true > Class Template Reference

```
#include <median.h>
```

Public Types

- typedef median_detail::result< STRSXP >::type result_type
- typedef Rcpp::traits::storage_type< STRSXP >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.399.1 Detailed Description

template<typename T>
class Rcpp::sugar::Median< STRSXP, false, T, true >

Definition at line 254 of file median.h.

6.399.2 Member Typedef Documentation

6.399.2.1 result_type

```
template<typename T >
typedef median_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, false, T, true >↩
::result_type
```

Definition at line 256 of file median.h.

6.399.2.2 stored_type

```
template<typename T >
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, false, T, true >↩
::stored_type
```

Definition at line 257 of file median.h.

6.399.2.3 VECTOR

```
template<typename T >
typedef T Rcpp::sugar::Median< STRSXP, false, T, true >::VECTOR
```

Definition at line 258 of file median.h.

6.399.3 Constructor & Destructor Documentation

6.399.3.1 Median()

```
template<typename T >
Rcpp::sugar::Median< STRSXP, false, T, true >::Median (
             const VECTOR & xx )  [inline]
```

Definition at line 264 of file median.h.

6.399.4 Member Function Documentation

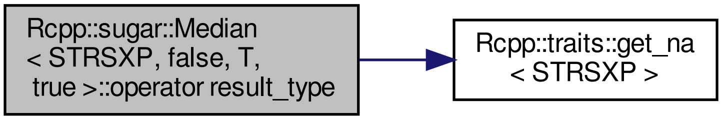
6.399.4.1 operator result_type()

```
template<typename T >
Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type ( )  [inline]
```

Definition at line 267 of file median.h.

References Rcpp::traits::get_na< STRSXP >(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.399.5 Member Data Documentation

6.399.5.1 x

```
template<typename T >
VECTOR Rcpp::sugar::Median< STRSXP, false, T, true >::x  [private]
```

Definition at line 261 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.400 Rcpp::sugar::Median< STRSXP, NA, T, NA_RM > Class Template Reference

```
#include <median.h>
```

Public Types

- typedef median_detail::result< STRSXP >::type result_type
- typedef Rcpp::traits::storage_type< STRSXP >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.400.1 Detailed Description

template<bool NA, typename T, bool NA_RM>
class Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >

Definition at line 196 of file median.h.

6.400.2 Member Typedef Documentation

6.400.2.1 result_type

```
template<bool NA, typename T , bool NA_RM>
typedef median_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::result_type
```

Definition at line 198 of file median.h.

6.400.2.2 stored_type

```
template<bool NA, typename T , bool NA_RM>
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >↩
::stored_type
```

Definition at line 199 of file median.h.

6.400.2.3 VECTOR

```
template<bool NA, typename T , bool NA_RM>
typedef T Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::VECTOR
```

Definition at line 200 of file median.h.

6.400.3 Constructor & Destructor Documentation

6.400.3.1 Median()

```
template<bool NA, typename T , bool NA_RM>
Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::Median (
             const VECTOR & xx )  [inline]
```

Definition at line 206 of file median.h.

6.400.4 Member Function Documentation

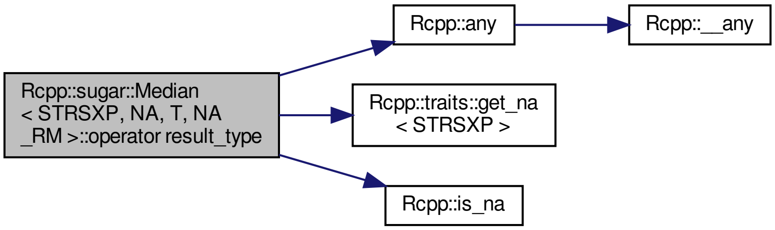
6.400.4.1 operator result_type()

```
template<bool NA, typename T , bool NA_RM>
Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type ( )  [inline]
```

Definition at line 209 of file median.h.

References Rcpp::any(), Rcpp::traits::get_na< STRSXP >(), Rcpp::is_na(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.400.5 Member Data Documentation

6.400.5.1 x

```
template<bool NA, typename T , bool NA_RM>
VECTOR Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::x  [private]
```

Definition at line 203 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.401 Rcpp::sugar::Median< STRSXP, NA, T, true > Class Template Reference

```
#include <median.h>
```

Public Types

- typedef median_detail::result< STRSXP >::type result_type
- typedef Rcpp::traits::storage_type< STRSXP >::type stored_type
- typedef T VECTOR

Public Member Functions

- Median (const VECTOR &xx)
- operator result_type ()

Private Attributes

- VECTOR x

6.401.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Median< STRSXP, NA, T, true >

Definition at line 227 of file median.h.

6.401.2 Member Typedef Documentation

6.401.2.1 result_type

```
template<bool NA, typename T >
typedef median_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, true >::result_type
```

Definition at line 229 of file median.h.

6.401.2.2 stored_type

```
template<bool NA, typename T >
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, true >↩
::stored_type
```

Definition at line 230 of file median.h.

6.401.2.3 VECTOR

```
template<bool NA, typename T >
typedef T Rcpp::sugar::Median< STRSXP, NA, T, true >::VECTOR
```

Definition at line 231 of file median.h.

6.401.3 Constructor & Destructor Documentation

6.401.3.1 Median()

```
template<bool NA, typename T >
Rcpp::sugar::Median< STRSXP, NA, T, true >::Median (
             const VECTOR & xx )  [inline]
```

Definition at line 237 of file median.h.

6.401.4 Member Function Documentation

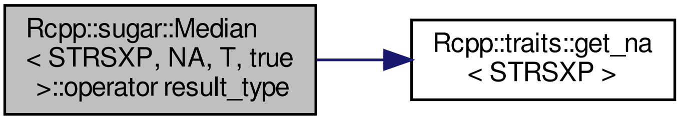
6.401.4.1 operator result_type()

```
template<bool NA, typename T >
Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result_type ( )  [inline]
```

Definition at line 240 of file median.h.

References Rcpp::traits::get_na< STRSXP >(), and Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x.

Here is the call graph for this function:



6.401.5 Member Data Documentation

6.401.5.1 x

```
template<bool NA, typename T >
VECTOR Rcpp::sugar::Median< STRSXP, NA, T, true >::x  [private]
```

Definition at line 234 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.402 Rcpp::sugar::Min< RTYPE, NA, T > Class Template Reference

```
#include <min.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Min (const T &obj_)
- operator STORAGE () const

Public Attributes

- const T & obj

6.402.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Min< RTYPE, NA, T >

Definition at line 28 of file min.h.

6.402.2 Member Typedef Documentation

6.402.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, NA, T >::STORAGE
```

Definition at line 30 of file min.h.

6.402.3 Constructor & Destructor Documentation

6.402.3.1 Min()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Min< RTYPE, NA, T >::Min (
             const T & obj_ ) [inline]
```

Definition at line 32 of file min.h.

6.402.4 Member Function Documentation

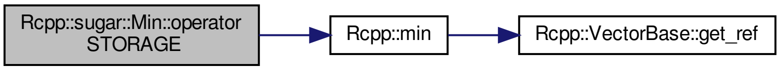
6.402.4.1 operator STORAGE()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE ( ) const  [inline]
```

Definition at line 34 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.402.5 Member Data Documentation

6.402.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Min< RTYPE, NA, T >::obj
```

Definition at line 50 of file min.h.

Referenced by Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE(), and Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/min.h

6.403 Rcpp::sugar::Min< RTYPE, false, T > Class Template Reference

```
#include <min.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Min (const T &obj_)
- operator STORAGE () const

Public Attributes

- const T & obj

6.403.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Min< RTYPE, false, T >

Definition at line 55 of file min.h.

6.403.2 Member Typedef Documentation

6.403.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, false, T >::STORAGE
```

Definition at line 57 of file min.h.

6.403.3 Constructor & Destructor Documentation

6.403.3.1 Min()

```
template<int RTYPE, typename T >
Rcpp::sugar::Min< RTYPE, false, T >::Min (
             const T & obj_ )  [inline]
```

Definition at line 59 of file min.h.

6.403.4 Member Function Documentation

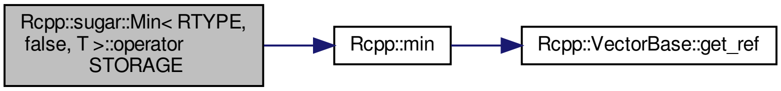
6.403.4.1 operator STORAGE()

```
template<int RTYPE, typename T >
Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE ( ) const  [inline]
```

Definition at line 61 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:


Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE
Rcpp::min
Rcpp::VectorBase::get_ref

6.403.5 Member Data Documentation

6.403.5.1 obj

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Min< RTYPE, false, T >::obj
```

Definition at line 74 of file min.h.

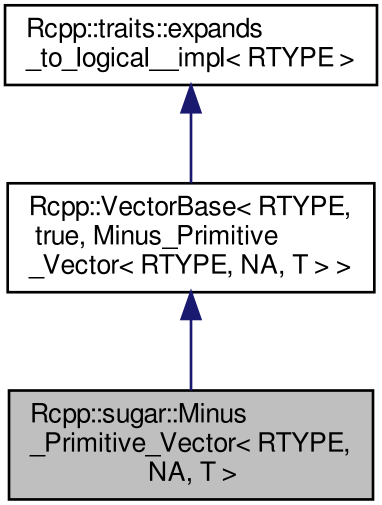
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/min.h

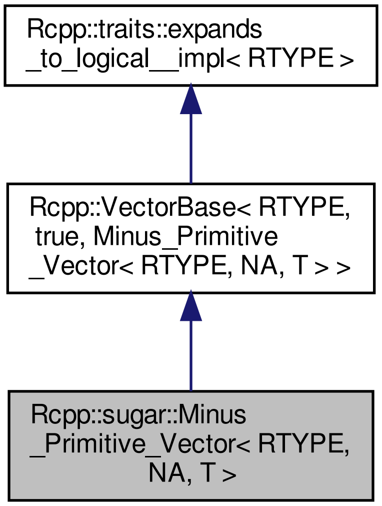
6.404 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, NA, T > >
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type VEC_EXT

Public Member Functions

- Minus_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- STORAGE lhs
- const VEC_EXT & rhs
- bool lhs_na

6.404.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >

Definition at line 324 of file minus.h.

6.404.2 Member Typedef Documentation

6.404.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >↩
::STORAGE
```

Definition at line 328 of file minus.h.

6.404.2.2 VEC_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA,
T >::VEC_EXT
```

Definition at line 329 of file minus.h.

6.404.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 327 of file minus.h.

6.404.3 Constructor & Destructor Documentation

6.404.3.1 Minus_Primitive_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::Minus_Primitive_Vector (
                STORAGE lhs_,
                const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 331 of file minus.h.

6.404.4 Member Function Documentation

6.404.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 334 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs, Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na, and Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.404.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 338 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.404.5 Member Data Documentation

6.404.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs  [private]
```

Definition at line 341 of file minus.h.

Referenced by Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::operator[ ]().

6.404.5.2 lhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na [private]
```

Definition at line 343 of file minus.h.

Referenced by Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[ ]().

6.404.5.3 rhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs [private]
```

Definition at line 342 of file minus.h.

Referenced by Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::size(), Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::size(), and Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::size().

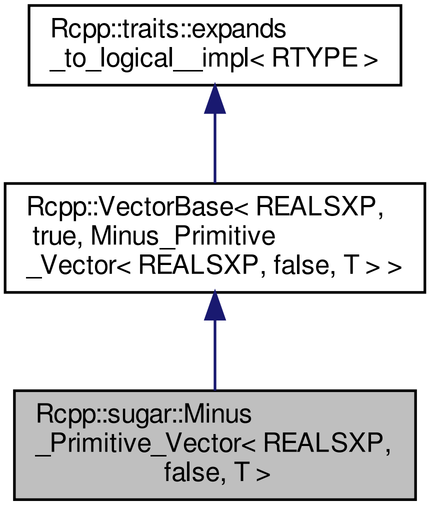
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

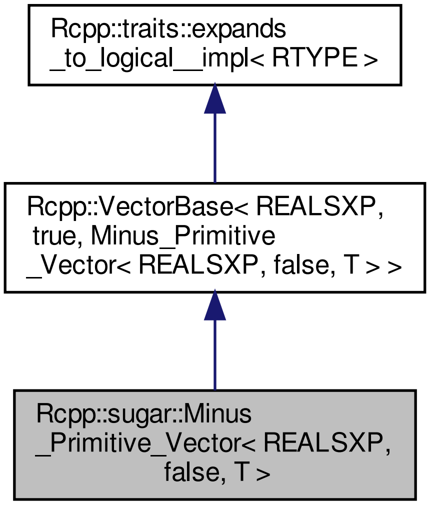
6.405 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type VEC_EXT

Public Member Functions

- Minus_Primitive_Vector (double lhs_, const VEC_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- double lhs
- const VEC_EXT & rhs

6.405.1 Detailed Description

template<typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >

Definition at line 392 of file minus.h.

6.405.2 Member Typedef Documentation

6.405.2.1 VEC_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Minus_Primitive_Vector< REALSXP,
false, T >::VEC_EXT
```

Definition at line 396 of file minus.h.

6.405.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 395 of file minus.h.

6.405.3 Constructor & Destructor Documentation

6.405.3.1 Minus_Primitive_Vector()

```
template<typename T >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::Minus_Primitive_Vector (
             double lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 398 of file minus.h.

6.405.4 Member Function Documentation

6.405.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 401 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.405.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 405 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.405.5 Member Data Documentation

6.405.5.1 lhs

```
template<typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::lhs  [private]
```

Definition at line 408 of file minus.h.

6.405.5.2 rhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::rhs  [private]
```

Definition at line 409 of file minus.h.

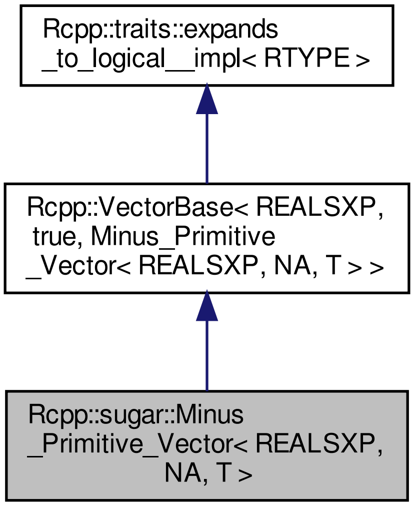
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

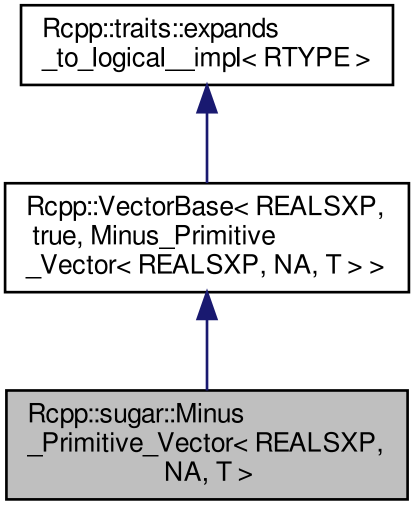
6.406 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Minus_Primitive
_Vector< REALSXP, NA, T > >
Rcpp::sugar::Minus
_Primitive_Vector< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Minus_Primitive _Vector< REALSXP, NA, T > >
Rcpp::sugar::Minus _Primitive_Vector< REALSXP, NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type VEC_EXT

Public Member Functions

- Minus_Primitive_Vector (double lhs_, const VEC_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- double lhs
- const VEC_EXT & rhs

6.406.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >

Definition at line 346 of file minus.h.

6.406.2 Member Typedef Documentation

6.406.2.1 VEC_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Minus_Primitive_Vector< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 350 of file minus.h.

6.406.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 349 of file minus.h.

6.406.3 Constructor & Destructor Documentation

6.406.3.1 Minus_Primitive_Vector()

```
template<bool NA, typename T >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::Minus_Primitive_Vector (
             double lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 352 of file minus.h.

6.406.4 Member Function Documentation

6.406.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 355 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.406.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 358 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.406.5 Member Data Documentation

6.406.5.1 lhs

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::lhs  [private]
```

Definition at line 361 of file minus.h.

6.406.5.2 rhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::rhs  [private]
```

Definition at line 362 of file minus.h.

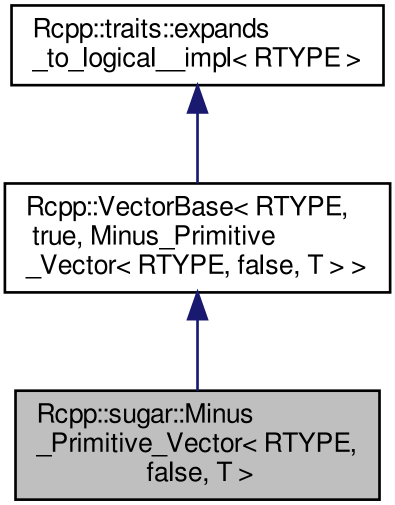
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

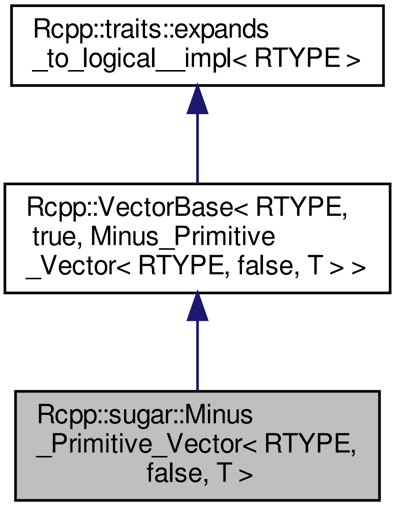
6.407 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type VEC_EXT

Public Member Functions

- Minus_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)
- STORAGE operator[] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- STORAGE lhs
- const VEC_EXT & rhs
- bool lhs_na

6.407.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >

Definition at line 368 of file minus.h.

6.407.2 Member Typedef Documentation

6.407.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T
>::STORAGE
```

Definition at line 372 of file minus.h.

6.407.2.2 VEC_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE,
false, T >::VEC_EXT
```

Definition at line 373 of file minus.h.

6.407.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >↩
::VEC_TYPE
```

Definition at line 371 of file minus.h.

6.407.3 Constructor & Destructor Documentation

6.407.3.1 Minus_Primitive_Vector()

```
template<int RTYPE, typename T >
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::Minus_Primitive_Vector (
             STORAGE lhs_,
             const VEC_TYPE & rhs_ )  [inline]
```

Definition at line 375 of file minus.h.

6.407.4 Member Function Documentation

6.407.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 378 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs, Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na, and Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.407.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 383 of file minus.h.

References Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs.

6.407.5 Member Data Documentation

6.407.5.1 lhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::lhs  [private]
```

Definition at line 386 of file minus.h.

6.407.5.2 lhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::lhs_na  [private]
```

Definition at line 388 of file minus.h.

6.407.5.3 rhs

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::rhs  [private]
```

Definition at line 387 of file minus.h.

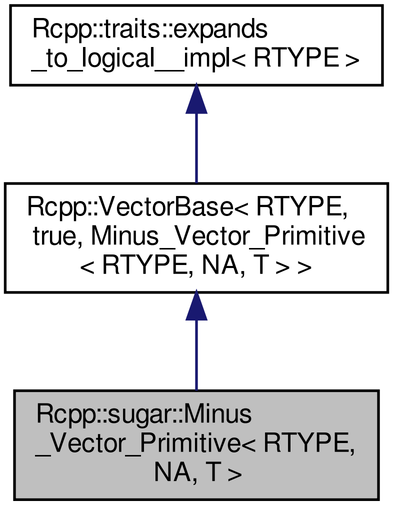
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

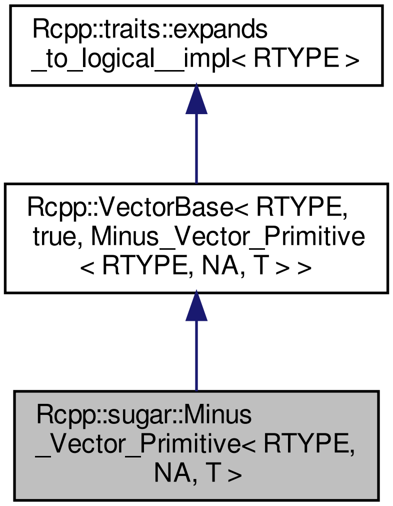
6.408 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >:



Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type VEC_EXT

Public Member Functions

- Minus_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- STORAGE rhs
- bool rhs_na

6.408.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >

Definition at line 228 of file minus.h.

6.408.2 Member Typedef Documentation

6.408.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >↩
::STORAGE
```

Definition at line 231 of file minus.h.

6.408.2.2 VEC_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA,
T >::VEC_EXT
```

Definition at line 233 of file minus.h.

6.408.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 232 of file minus.h.

6.408.3 Constructor & Destructor Documentation

6.408.3.1 Minus_Vector_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::Minus_Vector_Primitive (
             const VEC_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 235 of file minus.h.

6.408.4 Member Function Documentation

6.408.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 238 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.408.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 244 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.408.5 Member Data Documentation

6.408.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs  [private]
```

Definition at line 247 of file minus.h.

Referenced by Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[](), Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::operator[](), Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[](), Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::operator[](), Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::size(), Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::size(), Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::size(), and Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::size().

6.408.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs  [private]
```

Definition at line 248 of file minus.h.

Referenced by Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::operator[ ]().

6.408.5.3 rhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na  [private]
```

Definition at line 249 of file minus.h.

Referenced by Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[ ]().

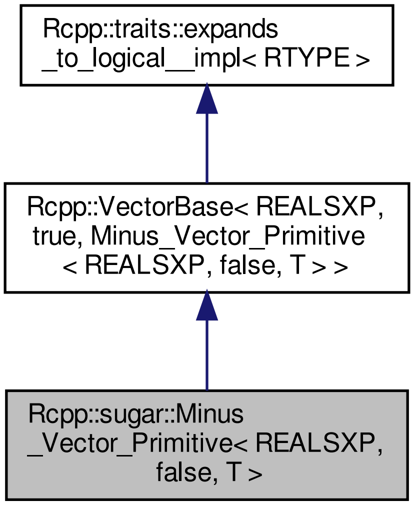
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

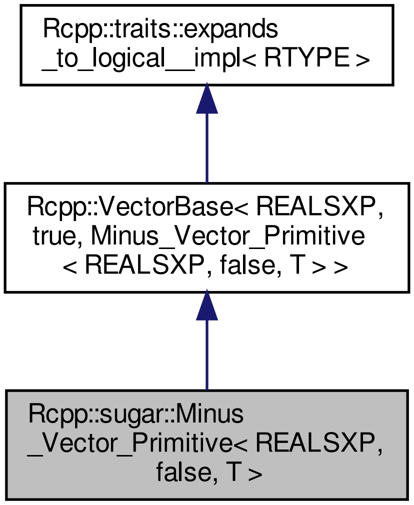
6.409 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Minus_Vector_Primitive
< REALSXP, false, T > >
Rcpp::sugar::Minus
_Vector_Primitive< REALSXP,
false, T >

Collaboration diagram for Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Minus_Vector_Primitive
< REALSXP, false, T > >
Rcpp::sugar::Minus
_Vector_Primitive< REALSXP,
false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type VEC_EXT

Public Member Functions

- Minus_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- double rhs

6.409.1 Detailed Description

template<typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >

Definition at line 298 of file minus.h.

6.409.2 Member Typedef Documentation

6.409.2.1 VEC_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Minus_Vector_Primitive< REALSXP,
false, T >::VEC_EXT
```

Definition at line 302 of file minus.h.

6.409.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 301 of file minus.h.

6.409.3 Constructor & Destructor Documentation

6.409.3.1 Minus_Vector_Primitive()

```
template<typename T >
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::Minus_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 304 of file minus.h.

6.409.4 Member Function Documentation

6.409.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 307 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs.

6.409.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 311 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.409.5 Member Data Documentation

6.409.5.1 lhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 314 of file minus.h.

6.409.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 315 of file minus.h.

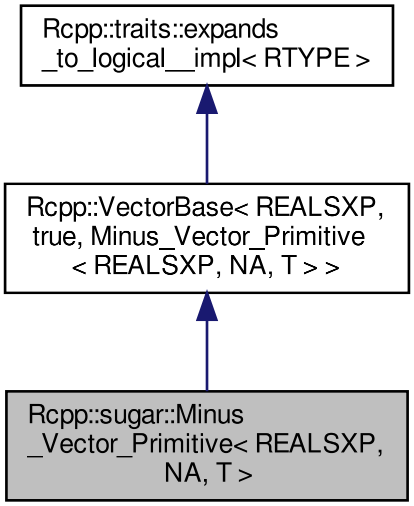
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

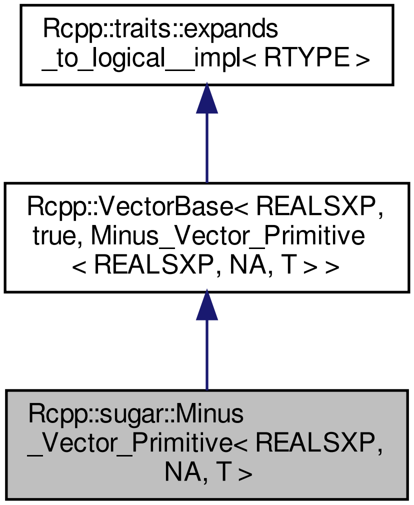
6.410 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Minus_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Minus
_Vector_Primitive< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Minus_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Minus
_Vector_Primitive< REALSXP,
NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type VEC_EXT

Public Member Functions

- Minus_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- double rhs

6.410.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >

Definition at line 252 of file minus.h.

6.410.2 Member Typedef Documentation

6.410.2.1 VEC_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Minus_Vector_Primitive< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 256 of file minus.h.

6.410.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 255 of file minus.h.

6.410.3 Constructor & Destructor Documentation

6.410.3.1 Minus_Vector_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::Minus_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 258 of file minus.h.

6.410.4 Member Function Documentation

6.410.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 261 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs.

6.410.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 265 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.410.5 Member Data Documentation

6.410.5.1 lhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::lhs [private]
```

Definition at line 268 of file minus.h.

6.410.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::rhs [private]
```

Definition at line 269 of file minus.h.

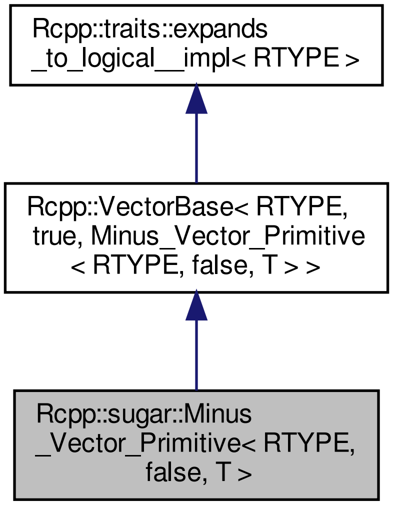
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

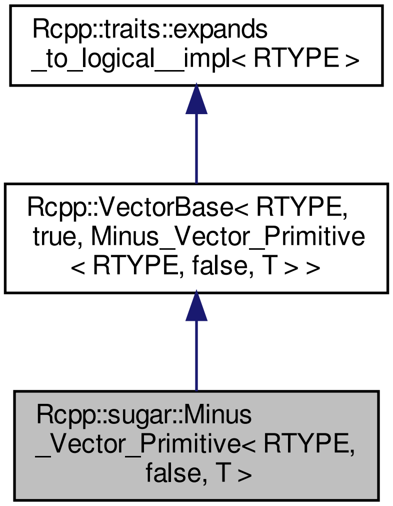
6.411 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >:



Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type VEC_EXT

Public Member Functions

- Minus_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & lhs
- STORAGE rhs
- bool rhs_na

6.411.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >

Definition at line 274 of file minus.h.

6.411.2 Member Typedef Documentation

6.411.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 277 of file minus.h.

6.411.2.2 VEC_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE,false,T>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE,
false, T >::VEC_EXT
```

Definition at line 279 of file minus.h.

6.411.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >↩
::VEC_TYPE
```

Definition at line 278 of file minus.h.

6.411.3 Constructor & Destructor Documentation

6.411.3.1 Minus_Vector_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::Minus_Vector_Primitive (
            const VEC_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 281 of file minus.h.

6.411.4 Member Function Documentation

6.411.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 284 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.411.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 290 of file minus.h.

References Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.411.5 Member Data Documentation

6.411.5.1 lhs

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::lhs  [private]
```

Definition at line 293 of file minus.h.

6.411.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::rhs  [private]
```

Definition at line 294 of file minus.h.

6.411.5.3 rhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::rhs_na  [private]
```

Definition at line 295 of file minus.h.

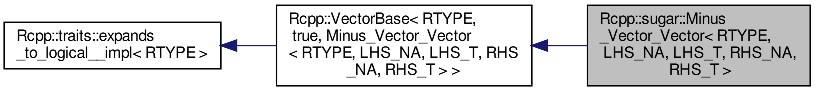
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

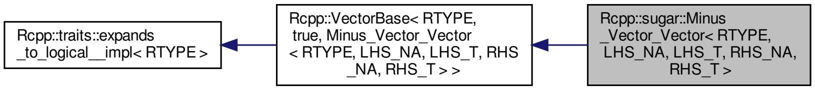
6.412 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.412.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file minus.h.

6.412.2 Member Typedef Documentation

6.412.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file minus.h.

6.412.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file minus.h.

6.412.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file minus.h.

6.412.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file minus.h.

6.412.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 34 of file minus.h.

6.412.3 Constructor & Destructor Documentation

6.412.3.1 Minus_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 38 of file minus.h.

6.412.4 Member Function Documentation

6.412.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 41 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.412.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.412.5 Member Data Documentation

6.412.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 51 of file minus.h.

Referenced by Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size().

6.412.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 52 of file minus.h.

Referenced by Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), and Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ]().

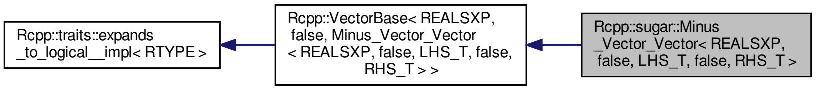
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

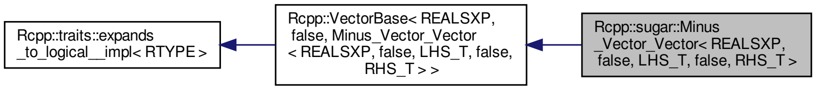
6.413 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.413.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Definition at line 201 of file minus.h.

6.413.2 Member Typedef Documentation

6.413.2.1 LHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 206 of file minus.h.

6.413.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 204 of file minus.h.

6.413.2.3 RHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 207 of file minus.h.

6.413.2.4 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 205 of file minus.h.

6.413.3 Constructor & Destructor Documentation

6.413.3.1 Minus_Vector_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 209 of file minus.h.

6.413.4 Member Function Documentation

6.413.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 212 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.413.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 216 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.413.5 Member Data Documentation

6.413.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 219 of file minus.h.

6.413.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 220 of file minus.h.

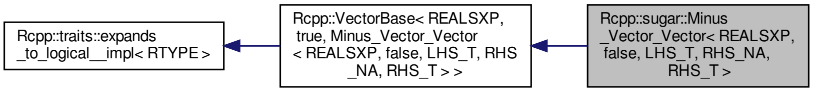
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

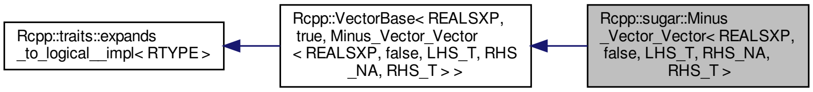
6.414 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector < REALSXP, false, LHS_T, RHS _NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector < REALSXP, false, LHS_T, RHS _NA, RHS_T > >
Rcpp::sugar::Minus _Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.414.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Definition at line 105 of file minus.h.

6.414.2 Member Typedef Documentation

6.414.2.1 LHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 110 of file minus.h.

6.414.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file minus.h.

6.414.2.3 RHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 111 of file minus.h.

6.414.2.4 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file minus.h.

6.414.3 Constructor & Destructor Documentation

6.414.3.1 Minus_Vector_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 113 of file minus.h.

6.414.4 Member Function Documentation

6.414.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 116 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.414.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 120 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.414.5 Member Data Documentation

6.414.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 123 of file minus.h.

6.414.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 124 of file minus.h.

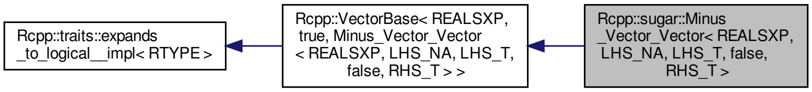
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

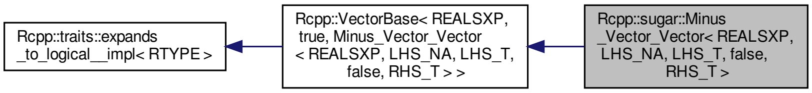
6.415 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.415.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Definition at line 154 of file minus.h.

6.415.2 Member Typedef Documentation

6.415.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 159 of file minus.h.

6.415.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 157 of file minus.h.

6.415.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 160 of file minus.h.

6.415.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 158 of file minus.h.

6.415.3 Constructor & Destructor Documentation

6.415.3.1 Minus_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 162 of file minus.h.

6.415.4 Member Function Documentation

6.415.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 165 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.415.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 169 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.415.5 Member Data Documentation

6.415.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 172 of file minus.h.

6.415.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 173 of file minus.h.

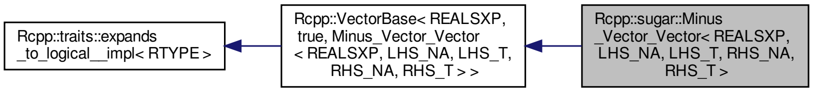
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

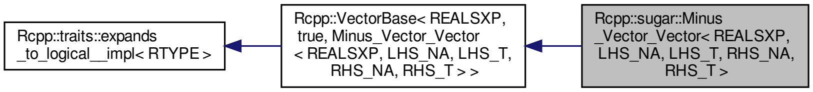
6.416 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.416.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 56 of file minus.h.

6.416.2 Member Typedef Documentation

6.416.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 61 of file minus.h.

6.416.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 59 of file minus.h.

6.416.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 62 of file minus.h.

6.416.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 60 of file minus.h.

6.416.3 Constructor & Destructor Documentation

6.416.3.1 Minus_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 64 of file minus.h.

6.416.4 Member Function Documentation

6.416.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 67 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.416.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 71 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.416.5 Member Data Documentation

6.416.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 74 of file minus.h.

6.416.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 75 of file minus.h.

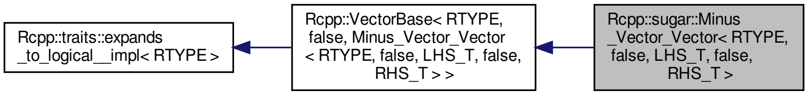
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

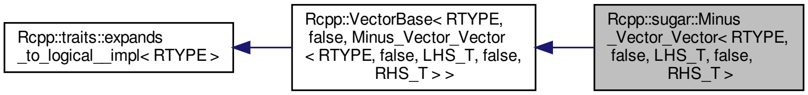
6.417 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.417.1 Detailed Description

template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Definition at line 178 of file minus.h.

6.417.2 Member Typedef Documentation

6.417.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 184 of file minus.h.

6.417.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS↩
_T, false, RHS_T >::LHS_TYPE
```

Definition at line 181 of file minus.h.

6.417.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 185 of file minus.h.

6.417.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS↩
_T, false, RHS_T >::RHS_TYPE
```

Definition at line 182 of file minus.h.

6.417.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 183 of file minus.h.

6.417.3 Constructor & Destructor Documentation

6.417.3.1 Minus_Vector_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Minus_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 187 of file minus.h.

6.417.4 Member Function Documentation

6.417.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 190 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.417.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 194 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.417.5 Member Data Documentation

6.417.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 197 of file minus.h.

6.417.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 198 of file minus.h.

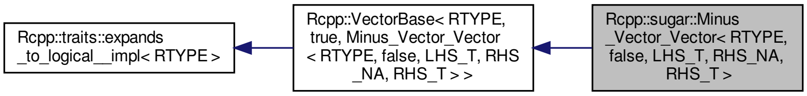
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

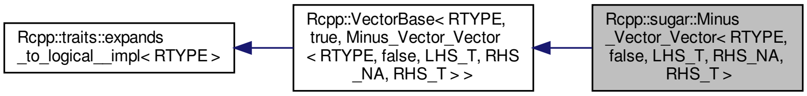
6.418 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:



Collaboration diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.418.1 Detailed Description

template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Definition at line 80 of file minus.h.

6.418.2 Member Typedef Documentation

6.418.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 85 of file minus.h.

6.418.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS↩
_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 82 of file minus.h.

6.418.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 86 of file minus.h.

6.418.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 83 of file minus.h.

6.418.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 84 of file minus.h.

6.418.3 Constructor & Destructor Documentation

6.418.3.1 Minus_Vector_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 88 of file minus.h.

6.418.4 Member Function Documentation

6.418.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 91 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.418.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 97 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.418.5 Member Data Documentation

6.418.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 100 of file minus.h.

6.418.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 101 of file minus.h.

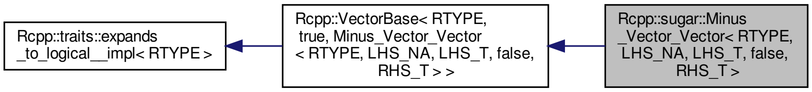
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

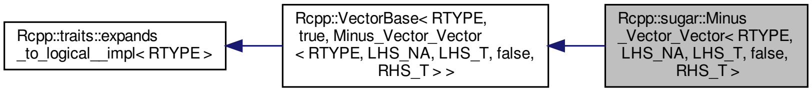
6.419 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Minus _Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Minus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.419.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Definition at line 129 of file minus.h.

6.419.2 Member Typedef Documentation

6.419.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 134 of file minus.h.

6.419.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 131 of file minus.h.

6.419.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 135 of file minus.h.

6.419.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 132 of file minus.h.

6.419.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 133 of file minus.h.

6.419.3 Constructor & Destructor Documentation

6.419.3.1 Minus_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Minus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 137 of file minus.h.

6.419.4 Member Function Documentation

6.419.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 140 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.419.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 146 of file minus.h.

References Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.419.5 Member Data Documentation

6.419.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 149 of file minus.h.

6.419.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 150 of file minus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

6.420 Rcpp::Module Class Reference

```
#include <Module.h>
```

Public Types

- typedef std::map< std::string, CppFunction ∗ > MAP
- typedef std::pair< const std::string, CppFunction ∗ > FUNCTION_PAIR
- typedef std::map< std::string, class_Base ∗ > CLASS_MAP
- typedef std::pair< const std::string, class_Base ∗ > CLASS_PAIR
- typedef CLASS_MAP::iterator CLASS_ITERATOR

Public Member Functions

- Module ()
- Module (const char ∗name_)
- SEXP invoke (const std::string &name_, SEXP ∗args, int nargs)
- IntegerVector functions_arity ()
- CharacterVector functions_names ()
- CharacterVector class_names ()
- List classes_info ()
- CharacterVector complete ()
- SEXP get_function (const std::string &name_)
- DL_FUNC get_function_ptr (const std::string &name_)
- void Add (const char ∗name_, 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

6.420.1 Detailed Description

holds information about exposed functions and classes

Definition at line 29 of file Module.h.

6.420.2 Member Typedef Documentation

6.420.2.1 CLASS_ITERATOR

```
typedef CLASS_MAP::iterator Rcpp::Module::CLASS_ITERATOR
```

Definition at line 36 of file Module.h.

6.420.2.2 CLASS_MAP

```
typedef std::map<std::string,class_Base*> Rcpp::Module::CLASS_MAP
```

Definition at line 34 of file Module.h.

6.420.2.3 CLASS_PAIR

```
typedef std::pair<const std::string,class_Base*> Rcpp::Module::CLASS_PAIR
```

Definition at line 35 of file Module.h.

6.420.2.4 FUNCTION_PAIR

```
typedef std::pair<const std::string,CppFunction*> Rcpp::Module::FUNCTION_PAIR
```

Definition at line 32 of file Module.h.

6.420.2.5 MAP

```
typedef std::map<std::string,CppFunction*> Rcpp::Module::MAP
```

Definition at line 31 of file Module.h.

6.420.3 Constructor & Destructor Documentation

6.420.3.1 Module() [1/2]

```
Rcpp::Module::Module ( )  [inline]
```

Definition at line 38 of file Module.h.

6.420.3.2 Module() [2/2]

```
Rcpp::Module::Module (
             const char * name_ )  [inline]
```

Definition at line 41 of file Module.h.

References name, and prefix.

6.420.4 Member Function Documentation

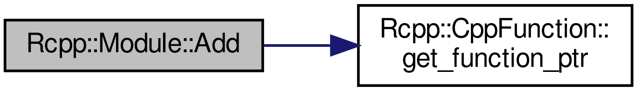
6.420.4.1 Add()

```
void Rcpp::Module::Add (
             const char * name_,
             CppFunction * ptr )  [inline]
```

Definition at line 192 of file Module.h.

References functions, Rcpp::CppFunction::get_function_ptr(), and prefix.

Here is the call graph for this function:



6.420.4.2 add_enum()

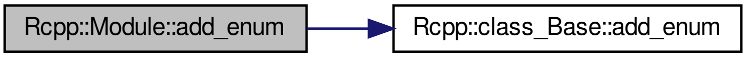
```
void Rcpp::Module::add_enum (
             const std::string & parent_class_typeinfo_name,
             const std::string & enum_name,
             const std::map< std::string, int > & value )  [inline]
```

Definition at line 219 of file Module.h.

References Rcpp::class_Base::add_enum(), and classes.

Referenced by Rcpp::enum_< Enum, Parent >::∼enum_().

Here is the call graph for this function:



6.420.4.3 AddClass()

```
void Rcpp::Module::AddClass (
             const char * name_,
             class_Base * cptr )  [inline]
```

Definition at line 197 of file Module.h.

References classes.

6.420.4.4 class_names()

```
CharacterVector Rcpp::Module::class_names ( )  [inline]
```

exposed class names

Definition at line 101 of file Module.h.

References classes.

6.420.4.5 classes_info()

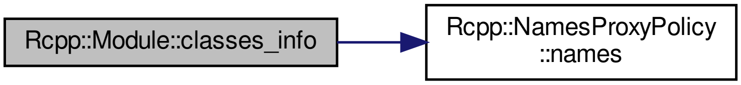
```
List Rcpp::Module::classes_info ( )  [inline]
```

information about the classes

Definition at line 25 of file Module.h.

References classes, and Rcpp::NamesProxyPolicy< CLASS >::names().

Here is the call graph for this function:



6.420.4.6 complete()

```
CharacterVector Rcpp::Module::complete ( )  [inline]
```

completion information

Definition at line 119 of file Module.h.

References classes, and functions.

6.420.4.7 functions_arity()

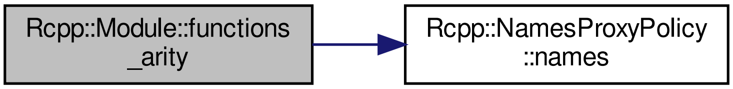
```
IntegerVector Rcpp::Module::functions_arity ( )  [inline]
```

vector of arity of all the functions exported by the module

Definition at line 72 of file Module.h.

References functions, and Rcpp::NamesProxyPolicy< CLASS >::names().

Here is the call graph for this function:


Rcpp::Module::functions _arity
Rcpp::NamesProxyPolicy ::names

6.420.4.8 functions_names()

```
CharacterVector Rcpp::Module::functions_names ( )  [inline]
```

vector of names of the functions

Definition at line 88 of file Module.h.

References functions.

6.420.4.9 get_class()

```
CppClass Rcpp::Module::get_class (
             const std::string & cl )  [inline]
```

Definition at line 39 of file Module.h.

References BEGIN_RCPP, classes, and END_RCPP.

6.420.4.10 get_class_pointer()

```
class_Base* Rcpp::Module::get_class_pointer (
             const std::string & cl )  [inline]
```

Definition at line 211 of file Module.h.

References classes.

6.420.4.11 get_function()

```
SEXP Rcpp::Module::get_function (
             const std::string & name_ )  [inline]
```

Returns a list that contains:

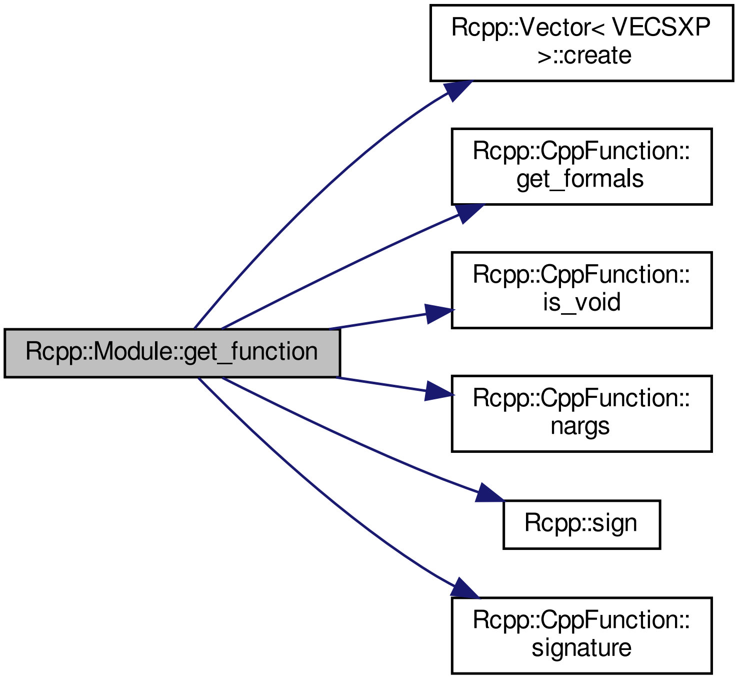
- an external pointer that encapsulates a CppFunction∗
- voidness of the function (logical)
- docstring (character)
- signature (character)
- formal arguments of the function

The R code in Module.R uses this information to create a C++Function object

Definition at line 154 of file Module.h.

References Rcpp::Vector< VECSXP >::create(), Rcpp::CppFunction::docstring, functions, Rcpp::CppFunction::get_formals(), Rcpp::CppFunction::is_void(), Rcpp::CppFunction::nargs(), Rcpp::sign(), and Rcpp::CppFunction::signature().

Here is the call graph for this function:


Rcpp::Module::get_function
Rcpp::Vector< VECSXP >::create
Rcpp::CppFunction:: get_formals
Rcpp::CppFunction:: is_void
Rcpp::CppFunction:: nargs
Rcpp::sign
Rcpp::CppFunction:: signature

6.420.4.12 get_function_ptr()

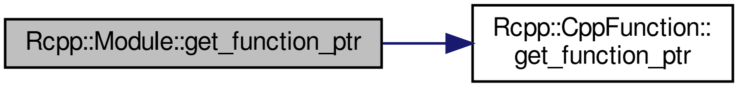
```
DL_FUNC Rcpp::Module::get_function_ptr (
             const std::string & name_ )  [inline]
```

get the underlying C++ function pointer as a DL_FUNC

Definition at line 179 of file Module.h.

References functions, and Rcpp::CppFunction::get_function_ptr().

Here is the call graph for this function:



6.420.4.13 has_class()

```
bool Rcpp::Module::has_class (
             const std::string & m )  [inline]
```

Definition at line 205 of file Module.h.

References classes.

6.420.4.14 has_function()

```
bool Rcpp::Module::has_function (
             const std::string & m )  [inline]
```

Definition at line 201 of file Module.h.

References functions.

6.420.4.15 invoke()

```
SEXP Rcpp::Module::invoke (
             const std::string & name_,
             SEXP * args,
             int nargs )  [inline]
```

calls a function from that module with the specified arguments

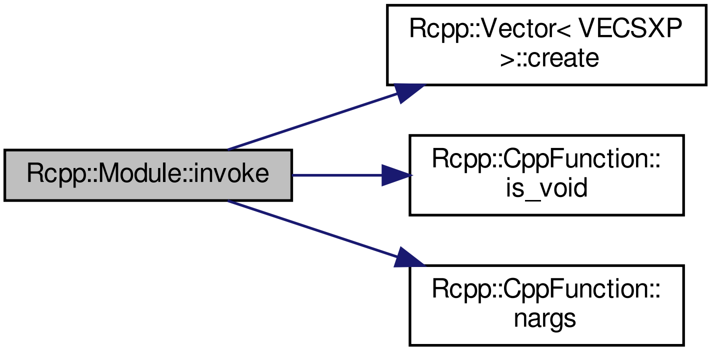
Parameters

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 53 of file Module.h.

References Rcpp::_, Rcpp::Vector< VECSXP >::create(), functions, Rcpp::CppFunction::is_void(), and Rcpp::Cpp↩ Function::nargs().

Here is the call graph for this function:


Rcpp::Vector< VECSXP >::create
Rcpp::Module::invoke
Rcpp::CppFunction:: is_void
Rcpp::CppFunction:: nargs

6.420.5 Member Data Documentation

6.420.5.1 classes

```
CLASS_MAP Rcpp::Module::classes  [private]
```

Definition at line 235 of file Module.h.

Referenced by add_enum(), AddClass(), class_names(), classes_info(), complete(), get_class(), get_class_pointer(), and has_class().

6.420.5.2 functions

```
MAP Rcpp::Module::functions  [private]
```

Definition at line 234 of file Module.h.

Referenced by Add(), complete(), functions_arity(), functions_names(), get_function(), get_function_ptr(), has_↩function(), and invoke().

6.420.5.3 name

```
std::string Rcpp::Module::name
```

Definition at line 217 of file Module.h.

Referenced by Module().

6.420.5.4 prefix

```
std::string Rcpp::Module::prefix  [private]
```

Definition at line 236 of file Module.h.

Referenced by Add(), and Module().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module.h

6.421 Rcpp::traits::module_wrap_traits< T > Struct Template Reference

```
#include <module_wrap_traits.h>
```

Public Types

- typedef normal_wrap_tag category

6.421.1 Detailed Description

template<typename T>
struct Rcpp::traits::module_wrap_traits< T >

Definition at line 33 of file module_wrap_traits.h.

6.421.2 Member Typedef Documentation

6.421.2.1 category

```
template<typename T >
typedef normal_wrap_tag Rcpp::traits::module_wrap_traits< T >::category
```

Definition at line 33 of file module_wrap_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

6.422 Rcpp::traits::module_wrap_traits< T * > Struct Template Reference

```
#include <module_wrap_traits.h>
```

Public Types

- typedef pointer_wrap_tag category

6.422.1 Detailed Description

template<typename T>
struct Rcpp::traits::module_wrap_traits< T * >

Definition at line 35 of file module_wrap_traits.h.

6.422.2 Member Typedef Documentation

6.422.2.1 category

```
template<typename T >
typedef pointer_wrap_tag Rcpp::traits::module_wrap_traits< T * >::category
```

Definition at line 35 of file module_wrap_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

6.423 Rcpp::traits::module_wrap_traits< void > Struct Reference

```
#include <module_wrap_traits.h>
```

Public Types

- typedef void_wrap_tag category

6.423.1 Detailed Description

Definition at line 34 of file module_wrap_traits.h.

6.423.2 Member Typedef Documentation

6.423.2.1 category

```
typedef void_wrap_tag Rcpp::traits::module_wrap_traits< void >::category
```

Definition at line 34 of file module_wrap_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

6.424 Rcpp::Na_Proxy Class Reference

```
#include <Na_Proxy.h>
```

Friends

- bool operator== (double x, Rcpp::Na_Proxy)
- bool operator== (int x, Rcpp::Na_Proxy)
- bool operator== (Rcpp::String x, Rcpp::Na_Proxy)
- bool operator== (Rcomplex x, Rcpp::Na_Proxy)
- bool operator== (SEXP x, Rcpp::Na_Proxy)
- bool operator== (std::string, Rcpp::Na_Proxy)
- bool operator== (const char ∗, Rcpp::Na_Proxy)
- bool operator== (Rcpp::internal::string_proxy< STRSXP > x, Rcpp::Na_Proxy)
- bool operator== (Rcpp::internal::const_string_proxy< STRSXP > x, Rcpp::Na_Proxy)
- bool operator== (Rcpp::Na_Proxy, double x)
- bool operator== (Rcpp::Na_Proxy, int x)
- bool operator== (Rcpp::Na_Proxy, Rcpp::String x)
- bool operator== (Rcpp::Na_Proxy, SEXP x)
- bool operator== (Rcpp::Na_Proxy, Rcomplex x)
- bool operator== (Rcpp::Na_Proxy, std::string)
- bool operator== (Rcpp::Na_Proxy, const char ∗)
- bool operator== (Rcpp::Na_Proxy, Rcpp::internal::string_proxy< STRSXP > x)
- bool operator== (Rcpp::Na_Proxy, Rcpp::internal::const_string_proxy< STRSXP > x)

6.424.1 Detailed Description

Definition at line 22 of file Na_Proxy.h.

6.424.2 Friends And Related Function Documentation

6.424.2.1 operator== [1/18]

```
bool operator== (
            const char * ,
            Rcpp::Na_Proxy  )  [friend]
```

Definition at line 30 of file Na_Proxy.h.

6.424.2.2 operator== [2/18]

```
bool operator== (
            double x,
            Rcpp::Na_Proxy  )  [friend]
```

Definition at line 24 of file Na_Proxy.h.

6.424.2.3 operator== [3/18]

```
bool operator== (
             int x,
             Rcpp::Na_Proxy  )  [friend]
```

Definition at line 25 of file Na_Proxy.h.

6.424.2.4 operator== [4/18]

```
bool operator== (
             Rcomplex x,
             Rcpp::Na_Proxy  )  [friend]
```

Definition at line 27 of file Na_Proxy.h.

6.424.2.5 operator== [5/18]

```
bool operator== (
             Rcpp::internal::const_string_proxy< STRSXP > x,
             Rcpp::Na_Proxy  )  [friend]
```

Definition at line 34 of file Na_Proxy.h.

6.424.2.6 operator== [6/18]

```
bool operator== (
             Rcpp::internal::string_proxy< STRSXP > x,
             Rcpp::Na_Proxy  )  [friend]
```

Definition at line 31 of file Na_Proxy.h.

6.424.2.7 operator== [7/18]

```
bool operator== (
             Rcpp::Na_Proxy ,
             const char *  )  [friend]
```

Definition at line 44 of file Na_Proxy.h.

6.424.2.8 operator== [8/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            double x )  [friend]
```

Definition at line 38 of file Na_Proxy.h.

6.424.2.9 operator== [9/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            int x )  [friend]
```

Definition at line 39 of file Na_Proxy.h.

6.424.2.10 operator== [10/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            Rcomplex x )  [friend]
```

Definition at line 42 of file Na_Proxy.h.

6.424.2.11 operator== [11/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            Rcpp::internal::const_string_proxy< STRSXP > x )  [friend]
```

Definition at line 48 of file Na_Proxy.h.

6.424.2.12 operator== [12/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            Rcpp::internal::string_proxy< STRSXP > x )  [friend]
```

Definition at line 45 of file Na_Proxy.h.

6.424.2.13 operator== [13/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            Rcpp::String x )  [friend]
```

Definition at line 40 of file Na_Proxy.h.

6.424.2.14 operator== [14/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            SEXP x )  [friend]
```

Definition at line 41 of file Na_Proxy.h.

6.424.2.15 operator== [15/18]

```
bool operator== (
            Rcpp::Na_Proxy ,
            std::string  )  [friend]
```

Definition at line 43 of file Na_Proxy.h.

6.424.2.16 operator== [16/18]

```
bool operator== (
            Rcpp::String x,
            Rcpp::Na_Proxy  )  [friend]
```

Definition at line 26 of file Na_Proxy.h.

6.424.2.17 operator== [17/18]

```
bool operator== (
            SEXP x,
            Rcpp::Na_Proxy  )  [friend]
```

Definition at line 28 of file Na_Proxy.h.

6.424.2.18 operator== [18/18]

```
bool operator== (
            std::string ,
            Rcpp::Na_Proxy  )  [friend]
```

Definition at line 29 of file Na_Proxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Na_Proxy.h

6.425 Rcpp::traits::named_object< T > Class Template Reference

```
#include <named_object.h>
```

Public Member Functions

- named_object (const std::string &name_, const T &o_)

Public Attributes

- const std::string & name
- const T & object

6.425.1 Detailed Description

template<typename T>
class Rcpp::traits::named_object< T >

Definition at line 33 of file named_object.h.

6.425.2 Constructor & Destructor Documentation

6.425.2.1 named_object()

```
template<typename T >
Rcpp::traits::named_object< T >::named_object (
            const std::string & name_,
            const T & o_ )  [inline]
```

Definition at line 35 of file named_object.h.

6.425.3 Member Data Documentation

6.425.3.1 name

```
template<typename T >
const std::string& Rcpp::traits::named_object< T >::name
```

Definition at line 37 of file named_object.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

6.425.3.2 object

```
template<typename T >
const T& Rcpp::traits::named_object< T >::object
```

Definition at line 38 of file named_object.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

6.426 Rcpp::traits::named_object< SEXP > Class Reference

```
#include <named_object.h>
```

Public Member Functions

- named_object (const std::string &name_, const SEXP &o_)
- named_object (const named_object< SEXP > &other)
- ∼named_object ()

Public Attributes

- const std::string & name
- SEXP object

Private Attributes

- SEXP token

6.426.1 Detailed Description

Definition at line 40 of file named_object.h.

6.426.2 Constructor & Destructor Documentation

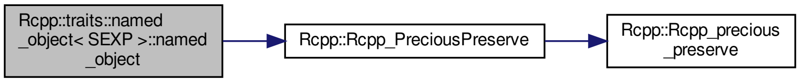
6.426.2.1 named_object() [1/2]

```
Rcpp::traits::named_object< SEXP >::named_object (
             const std::string & name_,
             const SEXP & o_ )  [inline]
```

Definition at line 42 of file named_object.h.

References Rcpp::Rcpp_PreciousPreserve().

Here is the call graph for this function:


Rcpp::traits::named_object< SEXP >::named_object
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve

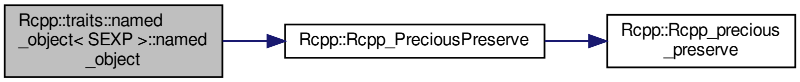
6.426.2.2 named_object() [2/2]

```
Rcpp::traits::named_object< SEXP >::named_object (
             const named_object< SEXP > & other )  [inline]
```

Definition at line 47 of file named_object.h.

References Rcpp::Rcpp_PreciousPreserve().

Here is the call graph for this function:


Rcpp::traits::named_object< SEXP >::named_object
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve

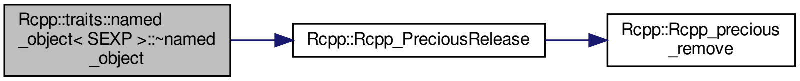
6.426.2.3 ∼named_object()

```
Rcpp::traits::named_object< SEXP >::∼named_object ( )  [inline]
```

Definition at line 51 of file named_object.h.

References Rcpp::Rcpp_PreciousRelease().

Here is the call graph for this function:



6.426.3 Member Data Documentation

6.426.3.1 name

```
const std::string& Rcpp::traits::named_object< SEXP >::name
```

Definition at line 55 of file named_object.h.

6.426.3.2 object

```
SEXP Rcpp::traits::named_object< SEXP >::object
```

Definition at line 56 of file named_object.h.

6.426.3.3 token

```
SEXP Rcpp::traits::named_object< SEXP >::token  [private]
```

Definition at line 58 of file named_object.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

6.427 Rcpp::internal::NamedPlaceHolder Class Reference

```
#include <Named.h>
```

Public Member Functions

- NamedPlaceHolder ()
- ∼NamedPlaceHolder ()
- Argument operator[ ] (const std::string &arg) const
- Argument operator() (const std::string &arg) const
- operator SEXP () const

6.427.1 Detailed Description

Definition at line 50 of file Named.h.

6.427.2 Constructor & Destructor Documentation

6.427.2.1 NamedPlaceHolder()

```
Rcpp::internal::NamedPlaceHolder::NamedPlaceHolder ( )  [inline]
```

Definition at line 52 of file Named.h.

6.427.2.2 ∼NamedPlaceHolder()

```
Rcpp::internal::NamedPlaceHolder::∼NamedPlaceHolder ( )  [inline]
```

Definition at line 53 of file Named.h.

6.427.3 Member Function Documentation

6.427.3.1 operator SEXP()

```
Rcpp::internal::NamedPlaceHolder::operator SEXP ( ) const  [inline]
```

Definition at line 60 of file Named.h.

6.427.3.2 operator()()

```
Argument Rcpp::internal::NamedPlaceHolder::operator() (
             const std::string & arg ) const  [inline]
```

Definition at line 57 of file Named.h.

6.427.3.3 operator[]()

```
Argument Rcpp::internal::NamedPlaceHolder::operator[] (
             const std::string & arg ) const  [inline]
```

Definition at line 54 of file Named.h.

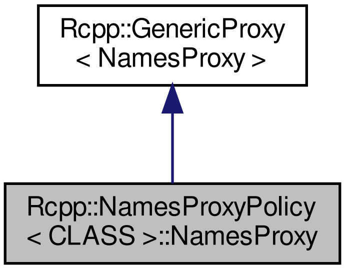
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Named.h

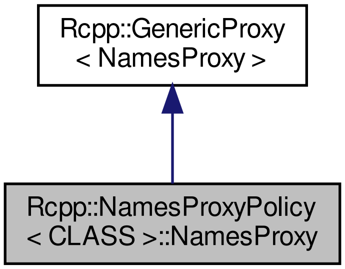
6.428 Rcpp::NamesProxyPolicy< CLASS >::NamesProxy Class Reference

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >::NamesProxy:



Collaboration diagram for Rcpp::NamesProxyPolicy< CLASS >::NamesProxy:



Public Member Functions

- NamesProxy (CLASS &v)
- NamesProxy & operator= (const NamesProxy &rhs)
- template< typename T >
  NamesProxy & operator= (const T &rhs)
- template< typename T >
  operator T () const
- template< typename T >
  NamesProxyPolicy< CLASS >::NamesProxy & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- CLASS & parent

6.428.1 Detailed Description

template< typename CLASS >
class Rcpp::NamesProxyPolicy< CLASS >::NamesProxy

Definition at line 27 of file NamesProxy.h.

6.428.2 Constructor & Destructor Documentation

6.428.2.1 NamesProxy()

```
template<typename CLASS >
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::NamesProxy (
             CLASS & v )  [inline]
```

Definition at line 29 of file NamesProxy.h.

6.428.3 Member Function Documentation

6.428.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get ( ) const  [inline], [private]
```

Definition at line 45 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent, and RCPP_GET_NAMES.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=().

6.428.3.2 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator T
```

Definition at line 70 of file proxy.h.

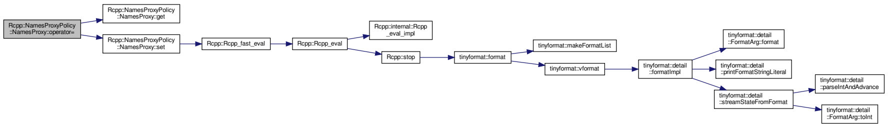
6.428.3.3 operator=() [1/3]

```
template<typename CLASS >
NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
             const NamesProxy & rhs )  [inline]
```

Definition at line 32 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get(), and Rcpp::NamesProxyPolicy< CLASS >::↩ NamesProxy::set().

Here is the call graph for this function:


Rcpp::NamesProxyPolicy ::NamesProxy::operator=
Rcpp::NamesProxyPolicy ::NamesProxy::get
Rcpp::NamesProxyPolicy ::NamesProxy::set
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

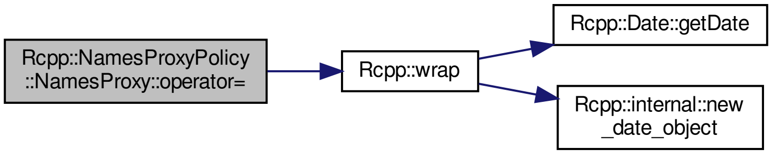
6.428.3.4 operator=() [2/3]

```
template<typename CLASS >
template<typename T >
NamesProxyPolicy<CLASS>::NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
             const T & rhs )
```

Definition at line 63 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::NamesProxyPolicy ::NamesProxy::operator=
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.428.3.5 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
              const T & rhs )
```

6.428.3.6 set()

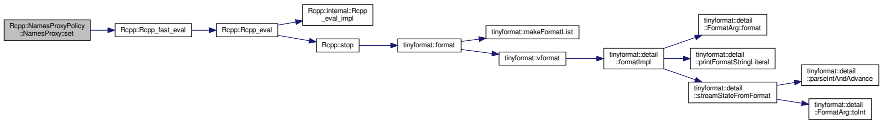
```
template<typename CLASS >
void Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set (
              SEXP x )  [inline], [private]
```

Definition at line 49 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent, and Rcpp::Rcpp_fast_eval().

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=().

Here is the call graph for this function:


Rcpp::NamesProxyPolicy ::NamesProxy::set
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.428.4 Member Data Documentation

6.428.4.1 parent

```
template<typename CLASS >
CLASS& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent  [private]
```

Definition at line 43 of file NamesProxy.h.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get(), and Rcpp::NamesProxyPolicy< CLASS >↩ ::NamesProxy::set().

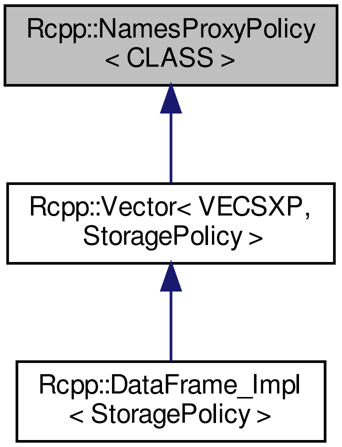
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/NamesProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.429 Rcpp::NamesProxyPolicy< CLASS > Class Template Reference

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >:


Rcpp::NamesProxyPolicy < CLASS >
Rcpp::Vector< VECSXP, StoragePolicy >
Rcpp::DataFrame_Impl < StoragePolicy >

Classes

- class const_NamesProxy
- class NamesProxy

Public Member Functions

- NamesProxy names ()
- const_NamesProxy names () const

6.429.1 Detailed Description

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >

Definition at line 24 of file NamesProxy.h.

6.429.2 Member Function Documentation

6.429.2.1 names() [1/2]

```
template<typename CLASS >
NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( )  [inline]
```

Definition at line 82 of file NamesProxy.h.

Referenced by Rcpp::Module::classes_info(), class_< Class >::fields(), Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), Rcpp::Module::functions_arity(), class_< Class >::getMethods(), lapplyCpp(), class_< Class >↩ ::methods_arity(), class_< Class >::methods_voidness(), and class_< Class >::property_classes().

6.429.2.2 names() [2/2]

```
template<typename CLASS >
const_NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( ) const  [inline]
```

Definition at line 86 of file NamesProxy.h.

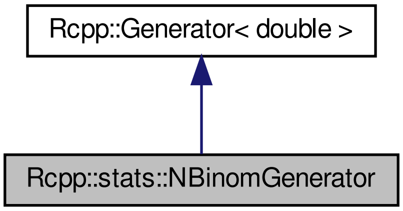
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/NamesProxy.h

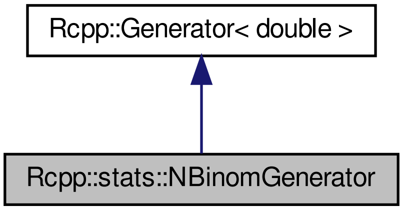
6.430 Rcpp::stats::NBinomGenerator Class Reference

```
#include <rnbinom.h>
```

Inheritance diagram for Rcpp::stats::NBinomGenerator:



Collaboration diagram for Rcpp::stats::NBinomGenerator:



Public Member Functions

- NBinomGenerator (double siz_, double prob_)
- double operator() () const

Private Attributes

- double siz
- double lambda

Additional Inherited Members

6.430.1 Detailed Description

Definition at line 28 of file rnbinom.h.

6.430.2 Constructor & Destructor Documentation

6.430.2.1 NBinomGenerator()

```
Rcpp::stats::NBinomGenerator::NBinomGenerator (
             double siz_,
             double prob_ )  [inline]
```

Definition at line 31 of file rnbinom.h.

6.430.3 Member Function Documentation

6.430.3.1 operator()()

```
double Rcpp::stats::NBinomGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rnbinom.h.

References lambda, and siz.

6.430.4 Member Data Documentation

6.430.4.1 lambda

```
double Rcpp::stats::NBinomGenerator::lambda  [private]
```

Definition at line 40 of file rnbinom.h.

Referenced by operator()().

6.430.4.2 siz

```
double Rcpp::stats::NBinomGenerator::siz  [private]
```

Definition at line 39 of file rnbinom.h.

Referenced by operator()().

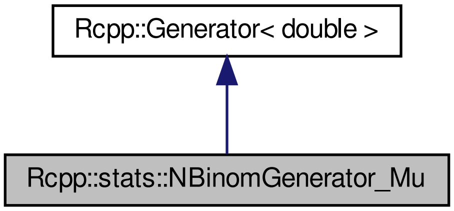
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnbinom.h

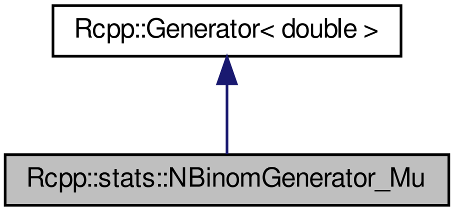
6.431 Rcpp::stats::NBinomGenerator_Mu Class Reference

```
#include <rnbinom_mu.h>
```

Inheritance diagram for Rcpp::stats::NBinomGenerator_Mu:



Collaboration diagram for Rcpp::stats::NBinomGenerator_Mu:



Public Member Functions

- NBinomGenerator_Mu (double siz_, double mu_)
- double operator() () const

Private Attributes

- double siz
- double lambda

Additional Inherited Members

6.431.1 Detailed Description

Definition at line 28 of file rnbinom_mu.h.

6.431.2 Constructor & Destructor Documentation

6.431.2.1 NBinomGenerator_Mu()

```
Rcpp::stats::NBinomGenerator_Mu::NBinomGenerator_Mu (
             double siz_,
             double mu_ )  [inline]
```

Definition at line 31 of file rnbinom_mu.h.

6.431.3 Member Function Documentation

6.431.3.1 operator()()

```
double Rcpp::stats::NBinomGenerator_Mu::operator() ( ) const  [inline]
```

Definition at line 34 of file rnbinom_mu.h.

References lambda, and siz.

6.431.4 Member Data Documentation

6.431.4.1 lambda

```
double Rcpp::stats::NBinomGenerator_Mu::lambda  [private]
```

Definition at line 40 of file rnbinom_mu.h.

Referenced by operator()().

6.431.4.2 siz

```
double Rcpp::stats::NBinomGenerator_Mu::siz  [private]
```

Definition at line 39 of file rnbinom_mu.h.

Referenced by operator()().

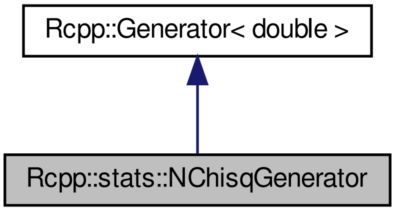
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnbinom_mu.h

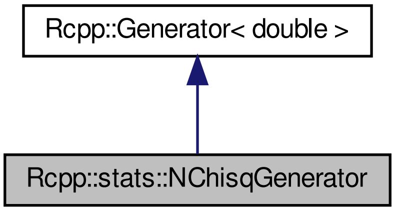
6.432 Rcpp::stats::NChisqGenerator Class Reference

```
#include <rnchisq.h>
```

Inheritance diagram for Rcpp::stats::NChisqGenerator:



Collaboration diagram for Rcpp::stats::NChisqGenerator:



Public Member Functions

- NChisqGenerator (double df_, double lambda_)
- double operator() () const

Private Attributes

- double df
- double df_2
- double lambda_2

Additional Inherited Members

6.432.1 Detailed Description

Definition at line 28 of file rnchisq.h.

6.432.2 Constructor & Destructor Documentation

6.432.2.1 NChisqGenerator()

```
Rcpp::stats::NChisqGenerator::NChisqGenerator (
              double df_,
              double lambda_ )  [inline]
```

Definition at line 31 of file rnchisq.h.

6.432.3 Member Function Documentation

6.432.3.1 operator()()

```
double Rcpp::stats::NChisqGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rnchisq.h.

References df, df_2, and lambda_2.

6.432.4 Member Data Documentation

6.432.4.1 df

```
double Rcpp::stats::NChisqGenerator::df  [private]
```

Definition at line 46 of file rnchisq.h.

Referenced by operator()().

6.432.4.2 df_2

```
double Rcpp::stats::NChisqGenerator::df_2  [private]
```

Definition at line 47 of file rnchisq.h.

Referenced by operator()().

6.432.4.3 lambda_2

```
double Rcpp::stats::NChisqGenerator::lambda_2  [private]
```

Definition at line 48 of file rnchisq.h.

Referenced by operator()().

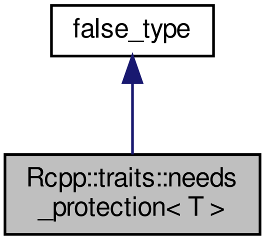
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnchisq.h

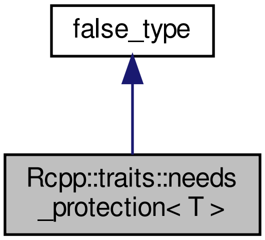
6.433 Rcpp::traits::needs_protection< T > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::needs_protection< T >:



Collaboration diagram for Rcpp::traits::needs_protection< T >:



Additional Inherited Members

6.433.1 Detailed Description

template<typename T>
struct Rcpp::traits::needs_protection< T >

Definition at line 30 of file named_object.h.

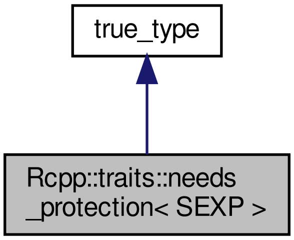
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

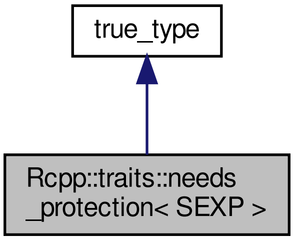
6.434 Rcpp::traits::needs_protection< SEXP > Struct Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::needs_protection< SEXP >:



Collaboration diagram for Rcpp::traits::needs_protection< SEXP >:



Additional Inherited Members

6.434.1 Detailed Description

Definition at line 31 of file named_object.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named_object.h

6.435 Rcpp::sugar::negate< NA > Struct Template Reference

```
#include <not.h>
```

Static Public Member Functions

- static int apply (int x)

6.435.1 Detailed Description

template<bool NA>
struct Rcpp::sugar::negate< NA >

Definition at line 28 of file not.h.

6.435.2 Member Function Documentation

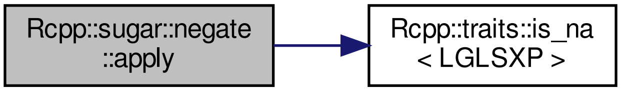
6.435.2.1 apply()

```
template<bool NA>
static int Rcpp::sugar::negate< NA >::apply (
             int x )  [inline], [static]
```

Definition at line 29 of file not.h.

References Rcpp::traits::is_na< LGLSXP >().

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/not.h

6.436 Rcpp::sugar::negate< false > Struct Reference

```
#include <not.h>
```

Static Public Member Functions

- static int apply (int x)

6.436.1 Detailed Description

Definition at line 34 of file not.h.

6.436.2 Member Function Documentation

6.436.2.1 apply()

```
static int Rcpp::sugar::negate< false >::apply (
                 int x )  [inline], [static]
```

Definition at line 35 of file not.h.

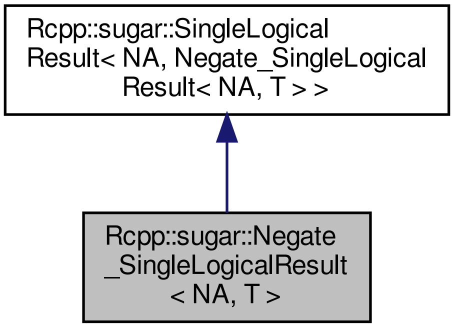
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/not.h

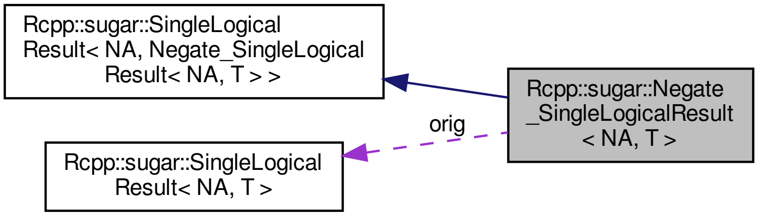
6.437 Rcpp::sugar::Negate_SingleLogicalResult< NA, T > Class Template Reference

```
#include <not.h>
```

Inheritance diagram for Rcpp::sugar::Negate_SingleLogicalResult< NA, T >:



Collaboration diagram for Rcpp::sugar::Negate_SingleLogicalResult< NA, T >:


Rcpp::sugar::SingleLogical
Result< NA, Negate_SingleLogical
Result< NA, T > >
Rcpp::sugar::Negate
_SingleLogicalResult
< NA, T >
orig
Rcpp::sugar::SingleLogical
Result< NA, T >

Public Types

- typedef SingleLogicalResult< NA, T > TYPE
- typedef SingleLogicalResult< NA, Negate_SingleLogicalResult< NA, T > > BASE

Public Member Functions

- Negate_SingleLogicalResult (const TYPE &orig_)
- void apply ()

Private Attributes

- const TYPE & orig

Additional Inherited Members

6.437.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Negate_SingleLogicalResult< NA, T >

Definition at line 42 of file not.h.

6.437.2 Member Typedef Documentation

6.437.2.1 BASE

```
template<bool NA, typename T >
typedef SingleLogicalResult<NA, Negate_SingleLogicalResult<NA,T> > Rcpp::sugar::Negate_SingleLogicalResult<
NA, T >::BASE
```

Definition at line 45 of file not.h.

6.437.2.2 TYPE

```
template<bool NA, typename T >
typedef SingleLogicalResult<NA,T> Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::TYPE
```

Definition at line 44 of file not.h.

6.437.3 Constructor & Destructor Documentation

6.437.3.1 Negate_SingleLogicalResult()

```
template<bool NA, typename T >
Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::Negate_SingleLogicalResult (
                const TYPE & orig_ )  [inline]
```

Definition at line 46 of file not.h.

6.437.4 Member Function Documentation

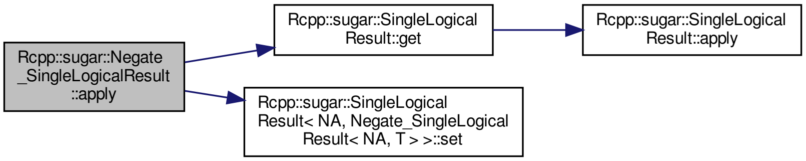
6.437.4.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply ( )  [inline]
```

Definition at line 48 of file not.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Negate_SingleLogicalResult< NA, T >↩ ::orig, and Rcpp::sugar::SingleLogicalResult< NA, Negate_SingleLogicalResult< NA, T > >::set().

Here is the call graph for this function:


Rcpp::sugar::Negate _SingleLogicalResult ::apply
Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::SingleLogical Result< NA, Negate_SingleLogical Result< NA, T > >::set

6.437.5 Member Data Documentation

6.437.5.1 orig

```
template<bool NA, typename T >
const TYPE& Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::orig  [private]
```

Definition at line 53 of file not.h.

Referenced by Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply().

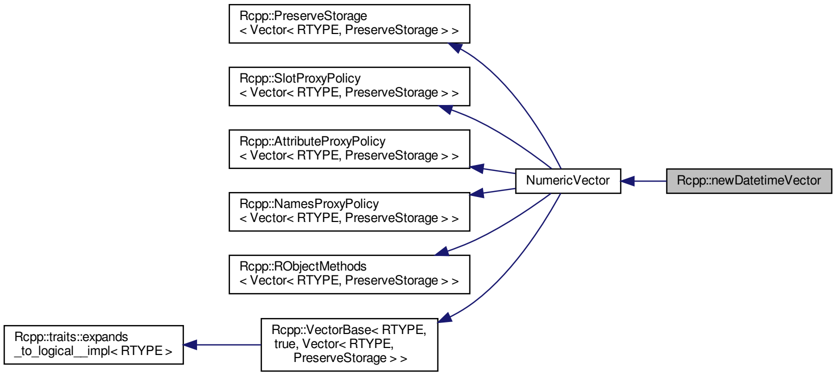
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/not.h

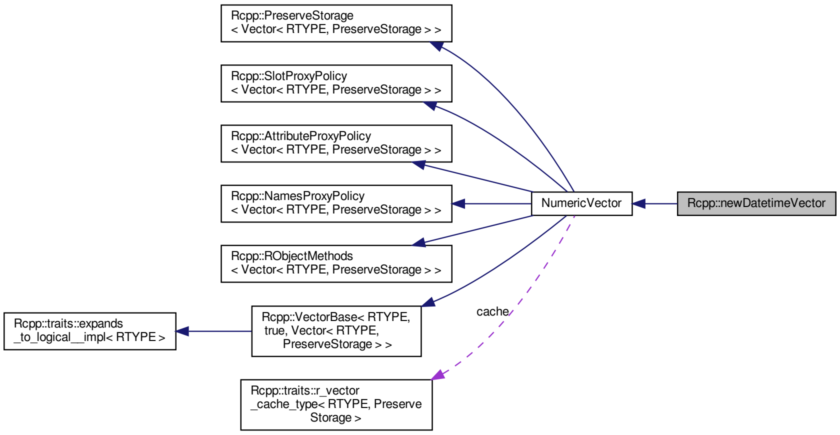
6.438 Rcpp::newDatetimeVector Class Reference

```
#include <newDatetimeVector.h>
```

Inheritance diagram for Rcpp::newDatetimeVector:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
NumericVector
Rcpp::newDatetimeVector

Collaboration diagram for Rcpp::newDatetimeVector:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
cache
Rcpp::traits::r_vector
_cache_type< RTYPE, Preserve
Storage >
NumericVector
Rcpp::newDatetimeVector

Public Member Functions

- template<int RTYPE, bool NA, typename VEC >
  newDatetimeVector (const VectorBase< RTYPE, NA, VEC > &other, const char ∗tz="")
- newDatetimeVector (SEXP vec, const char ∗tz="")
- newDatetimeVector (int n, const char ∗tz="")
- std::vector< Datetime > getDatetimes () const
- newDatetimeVector & operator= (const newDatetimeVector &rhs)

Private Member Functions

- void setClass (const char ∗tz)

Friends

- std::ostream & operator<< (std::ostream &s, const newDatetimeVector d)

Additional Inherited Members

6.438.1 Detailed Description

Definition at line 29 of file newDatetimeVector.h.

6.438.2 Constructor & Destructor Documentation

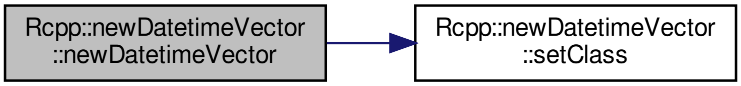
6.438.2.1 newDatetimeVector() [1/3]

```
template<int RTYPE, bool NA, typename VEC >
Rcpp::newDatetimeVector::newDatetimeVector (
            const VectorBase< RTYPE, NA, VEC > & other,
            const char * tz = "" )  [inline]
```

Definition at line 32 of file newDatetimeVector.h.

References setClass().

Here is the call graph for this function:



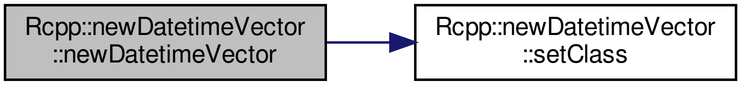
6.438.2.2 newDatetimeVector() [2/3]

```
Rcpp::newDatetimeVector::newDatetimeVector (
             SEXP vec,
             const char * tz = "" )  [inline]
```

Definition at line 37 of file newDatetimeVector.h.

References setClass().

Here is the call graph for this function:



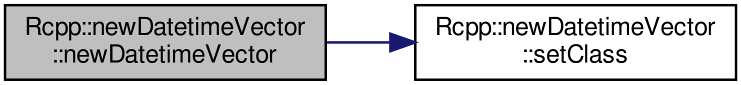
6.438.2.3 newDatetimeVector() [3/3]

```
Rcpp::newDatetimeVector::newDatetimeVector (
             int n,
             const char * tz = "" )  [inline]
```

Definition at line 42 of file newDatetimeVector.h.

References setClass().

Here is the call graph for this function:



6.438.3 Member Function Documentation

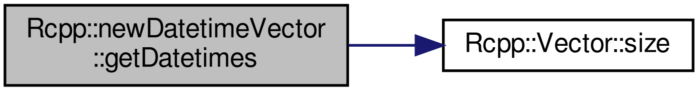
6.438.3.1 getDatetimes()

```
std::vector<Datetime> Rcpp::newDatetimeVector::getDatetimes ( ) const  [inline]
```

Definition at line 47 of file newDatetimeVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



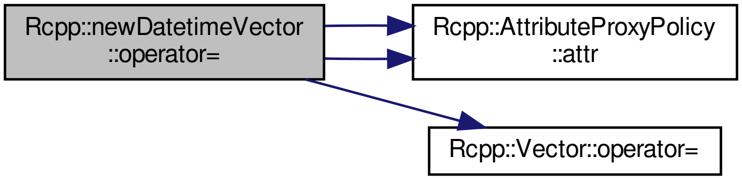
6.438.3.2 operator=()

```
newDatetimeVector& Rcpp::newDatetimeVector::operator= (
             const newDatetimeVector & rhs )  [inline]
```

Definition at line 55 of file newDatetimeVector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::AttributeProxyPolicy< Vector< RTYPE, Preserve↵Storage > >::attr(), and Rcpp::Vector< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



6.438.3.3 setClass()

```
void Rcpp::newDatetimeVector::setClass (
             const char * tz )  [inline], [private]
```

Definition at line 68 of file newDatetimeVector.h.

Referenced by newDatetimeVector().

6.438.4 Friends And Related Function Documentation

6.438.4.1 operator<<

```
std::ostream& operator<< (
             std::ostream & s,
             const newDatetimeVector d )  [friend]
```

Definition at line 81 of file newDatetimeVector.h.

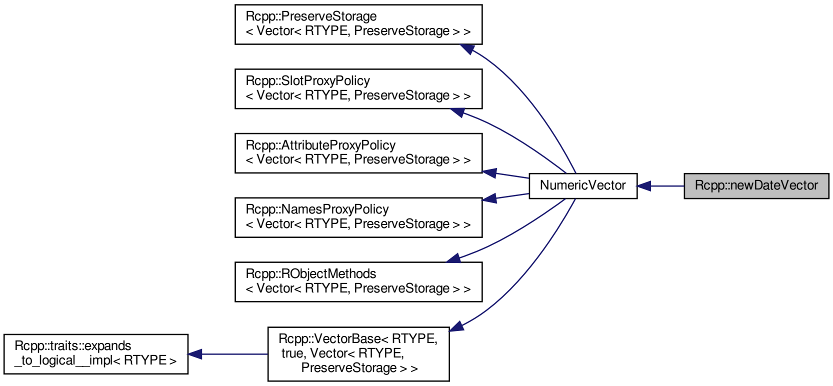
The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/newDatetimeVector.h

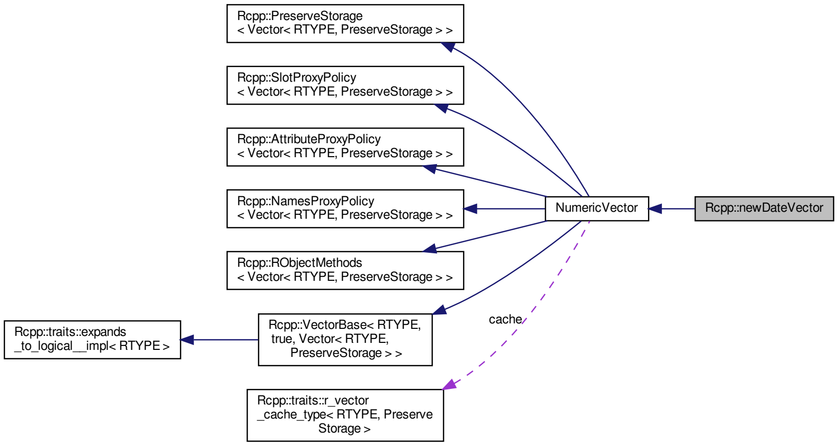
6.439 Rcpp::newDateVector Class Reference

```
#include <newDateVector.h>
```

Inheritance diagram for Rcpp::newDateVector:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
NumericVector
Rcpp::newDateVector

Collaboration diagram for Rcpp::newDateVector:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
NumericVector
Rcpp::newDateVector
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
cache
Rcpp::traits::r_vector
_cache_type< RTYPE, Preserve
Storage >

Public Member Functions

- template<int RTYPE, bool NA, typename VEC >
  newDateVector (const VectorBase< RTYPE, NA, VEC > &vec)
- newDateVector (SEXP vec)
- newDateVector (int n)
- std::vector< Date > getDates () const
- newDateVector & operator= (const newDateVector &rhs)

Private Member Functions

- void setClass ()

Friends

- std::ostream & operator<< (std::ostream &s, const newDateVector d)

Additional Inherited Members

6.439.1 Detailed Description

Definition at line 29 of file newDateVector.h.

6.439.2 Constructor & Destructor Documentation

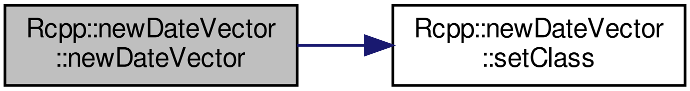
6.439.2.1 newDateVector() [1/3]

```
template<int RTYPE, bool NA, typename VEC >
Rcpp::newDateVector::newDateVector (
             const VectorBase< RTYPE, NA, VEC > & vec )  [inline]
```

Definition at line 32 of file newDateVector.h.

References setClass().

Here is the call graph for this function:


Rcpp::newDateVector ::newDateVector
Rcpp::newDateVector ::setClass

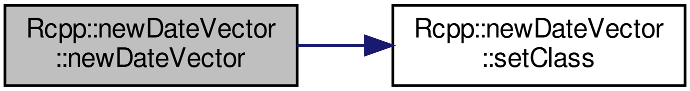
6.439.2.2 newDateVector() [2/3]

```
Rcpp::newDateVector::newDateVector (
             SEXP vec )  [inline]
```

Definition at line 36 of file newDateVector.h.

References setClass().

Here is the call graph for this function:


Rcpp::newDateVector ::newDateVector
Rcpp::newDateVector ::setClass

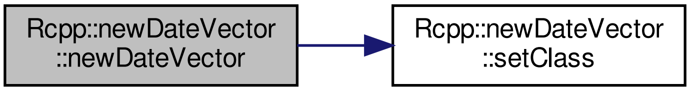
6.439.2.3 newDateVector() [3/3]

```
Rcpp::newDateVector::newDateVector (
             int n )  [inline]
```

Definition at line 37 of file newDateVector.h.

References setClass().

Here is the call graph for this function:


Rcpp::newDateVector ::newDateVector
Rcpp::newDateVector ::setClass

6.439.3 Member Function Documentation

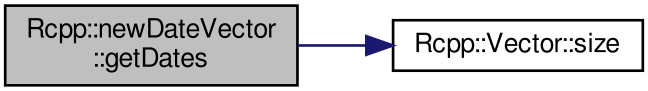
6.439.3.1 getDates()

```
std::vector<Date> Rcpp::newDateVector::getDates ( ) const  [inline]
```

Definition at line 39 of file newDateVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



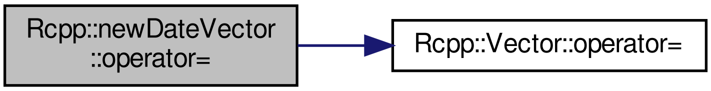
6.439.3.2 operator=()

```
newDateVector& Rcpp::newDateVector::operator= (
             const newDateVector & rhs )  [inline]
```

Definition at line 47 of file newDateVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



6.439.3.3 setClass()

```
void Rcpp::newDateVector::setClass ( )  [inline], [private]
```

Definition at line 59 of file newDateVector.h.

Referenced by newDateVector().

6.439.4 Friends And Related Function Documentation

6.439.4.1 operator<<

```
std::ostream& operator<< (
             std::ostream & s,
             const newDateVector d )  [friend]
```

Definition at line 65 of file newDateVector.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/newDateVector.h

6.440 Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no Struct Reference

Public Attributes

- char array [2]

6.440.1 Detailed Description

template<typename T>
struct Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no

Definition at line 376 of file cbind.h.

6.440.2 Member Data Documentation

6.440.2.1 array

```
template<typename T >
char Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no::array[2]
```

Definition at line 377 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

6.441 Rcpp::no_init_matrix Class Reference

```
#include <no_init.h>
```

Public Member Functions

- no_init_matrix (int nr_, int nc_)
- int nrow () const
- int ncol () const
- template<int RTYPE, template< class > class StoragePolicy>
  operator Matrix< RTYPE, StoragePolicy > () const

Private Attributes

- int nr
- int nc

6.441.1 Detailed Description

Definition at line 51 of file no_init.h.

6.441.2 Constructor & Destructor Documentation

6.441.2.1 no_init_matrix()

```
Rcpp::no_init_matrix::no_init_matrix (
             int nr_,
             int nc_ )  [inline]
```

Definition at line 53 of file no_init.h.

6.441.3 Member Function Documentation

6.441.3.1 ncol()

```
int Rcpp::no_init_matrix::ncol ( ) const  [inline]
```

Definition at line 59 of file no_init.h.

References nc.

6.441.3.2 nrow()

```
int Rcpp::no_init_matrix::nrow ( ) const  [inline]
```

Definition at line 55 of file no_init.h.

References nr.

6.441.3.3 operator Matrix< RTYPE, StoragePolicy >()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::no_init_matrix::operator Matrix< RTYPE, StoragePolicy > ( ) const  [inline]
```

Definition at line 64 of file no_init.h.

References nc, and nr.

6.441.4 Member Data Documentation

6.441.4.1 nc

```
int Rcpp::no_init_matrix::nc  [private]
```

Definition at line 74 of file no_init.h.

Referenced by ncol(), and operator Matrix< RTYPE, StoragePolicy >().

6.441.4.2 nr

```
int Rcpp::no_init_matrix::nr  [private]
```

Definition at line 73 of file no_init.h.

Referenced by nrow(), and operator Matrix< RTYPE, StoragePolicy >().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/no_init.h

6.442 Rcpp::no_init_vector Class Reference

```
#include <no_init.h>
```

Public Member Functions

- no_init_vector (R_xlen_t size_)
- R_xlen_t get () const
- template<int RTYPE, template< class > class StoragePolicy>
  operator Vector< RTYPE, StoragePolicy > () const

Private Attributes

- R_xlen_t size

6.442.1 Detailed Description

Definition at line 30 of file no_init.h.

6.442.2 Constructor & Destructor Documentation

6.442.2.1 no_init_vector()

```
Rcpp::no_init_vector::no_init_vector (
            R_xlen_t size_ )  [inline]
```

Definition at line 32 of file no_init.h.

6.442.3 Member Function Documentation

6.442.3.1 get()

```
R_xlen_t Rcpp::no_init_vector::get ( ) const  [inline]
```

Definition at line 34 of file no_init.h.

References size.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

6.442.3.2 operator Vector< RTYPE, StoragePolicy >()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::no_init_vector::operator Vector< RTYPE, StoragePolicy > ( ) const  [inline]
```

Definition at line 39 of file no_init.h.

References size.

6.442.4 Member Data Documentation

6.442.4.1 size

```
R_xlen_t Rcpp::no_init_vector::size  [private]
```

Definition at line 48 of file no_init.h.

Referenced by get(), and operator Vector< RTYPE, StoragePolicy >().

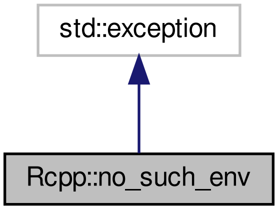
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/no_init.h

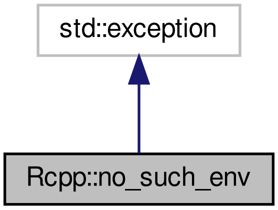
6.443 Rcpp::no_such_env Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::no_such_env:



Collaboration diagram for Rcpp::no_such_env:



Public Member Functions

- no_such_env (const std::string &name) throw ()
- no_such_env (int pos) throw ()
- virtual ∼no_such_env () throw ()
- virtual const char ∗ what () const throw ()

Private Attributes

- std::string message

6.443.1 Detailed Description

Definition at line 70 of file exceptions.h.

6.443.2 Constructor & Destructor Documentation

6.443.2.1 no_such_env() [1/2]

```
Rcpp::no_such_env::no_such_env (
            const std::string & name ) throw ( )   [inline]
```

Definition at line 72 of file exceptions.h.

6.443.2.2 no_such_env() [2/2]

```
Rcpp::no_such_env::no_such_env (
            int pos ) throw ( )   [inline]
```

Definition at line 74 of file exceptions.h.

6.443.2.3 ∼no_such_env()

```
virtual Rcpp::no_such_env::∼no_such_env ( ) throw ( )   [inline], [virtual]
```

Definition at line 76 of file exceptions.h.

6.443.3 Member Function Documentation

6.443.3.1 what()

```
virtual const char* Rcpp::no_such_env::what ( ) const throw ( )   [inline], [virtual]
```

Definition at line 77 of file exceptions.h.

References message.

6.443.4 Member Data Documentation

6.443.4.1 message

```
std::string Rcpp::no_such_env::message  [private]
```

Definition at line 79 of file exceptions.h.

Referenced by what().

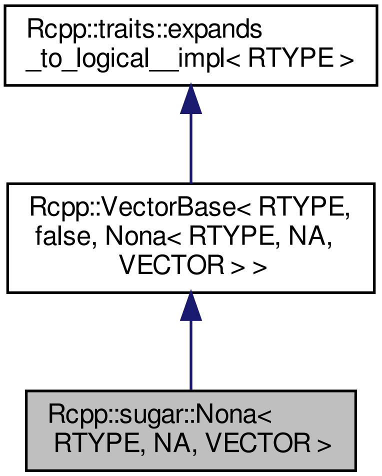
The documentation for this class was generated from the following file:

- inst/include/Rcpp/exceptions.h

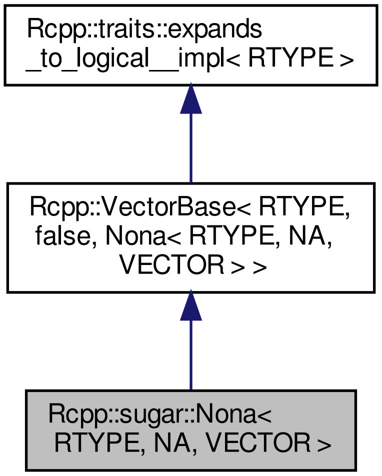
6.444 Rcpp::sugar::Nona< RTYPE, NA, VECTOR > Class Template Reference

```
#include <nona.h>
```

Inheritance diagram for Rcpp::sugar::Nona< RTYPE, NA, VECTOR >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, VECTOR > >
Rcpp::sugar::Nona< RTYPE, NA, VECTOR >

Collaboration diagram for Rcpp::sugar::Nona< RTYPE, NA, VECTOR >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, VECTOR > >
Rcpp::sugar::Nona< RTYPE, NA, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, VECTOR > SUGAR_TYPE

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Nona (const SUGAR_TYPE &expr)
- R_xlen_t size () const
- STORAGE operator[ ] (R_xlen_t i) const

Private Attributes

- const VECTOR & data

6.444.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
class Rcpp::sugar::Nona< RTYPE, NA, VECTOR >

Definition at line 29 of file nona.h.

6.444.2 Member Typedef Documentation

6.444.2.1 STORAGE

```
template<int RTYPE, bool NA, typename VECTOR >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::STORAGE
```

Definition at line 32 of file nona.h.

6.444.2.2 SUGAR_TYPE

```
template<int RTYPE, bool NA, typename VECTOR >
typedef Rcpp::VectorBase<RTYPE,NA,VECTOR> Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::SUGAR_TYPE
```

Definition at line 31 of file nona.h.

6.444.3 Constructor & Destructor Documentation

6.444.3.1 Nona()

```
template<int RTYPE, bool NA, typename VECTOR >
Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::Nona (
             const SUGAR_TYPE & expr )  [inline]
```

Definition at line 34 of file nona.h.

6.444.4 Member Function Documentation

6.444.4.1 operator[]()

```
template<int RTYPE, bool NA, typename VECTOR >
STORAGE Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

6.444.4.2 size()

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::size ( ) const  [inline]
```

Definition at line 36 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

6.444.5 Member Data Documentation

6.444.5.1 data

```
template<int RTYPE, bool NA, typename VECTOR >
const VECTOR& Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data  [private]
```

Definition at line 40 of file nona.h.

Referenced by Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::operator[ ](), Rcpp::sugar::Nona< RTYPE, NA, Rcpp↩ ::Vector< RTYPE > >::operator[ ](), and Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::size().

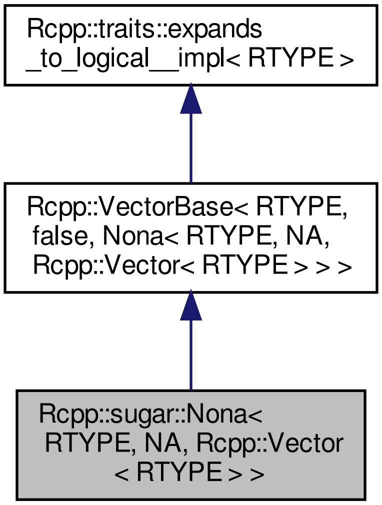
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

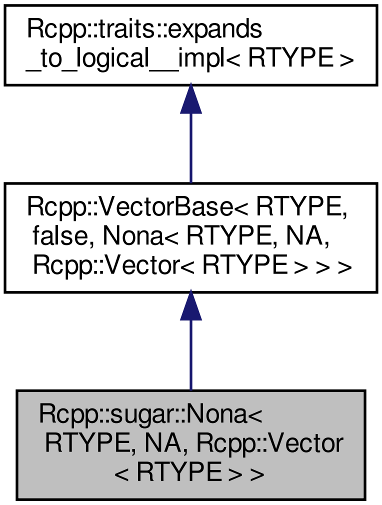
6.445 Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > Class Template Reference

```
#include <nona.h>
```

Inheritance diagram for Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > >
Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector < RTYPE > >

Collaboration diagram for Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, Rcpp::Vector< RTYPE > > SUGAR_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::Vector< RTYPE >::const_iterator iterator

Public Member Functions

- Nona (const SUGAR_TYPE &expr)
- R_xlen_t size () const
- STORAGE operator[ ] (R_xlen_t i) const

Private Attributes

- iterator data
- R_xlen_t n

6.445.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >

Definition at line 45 of file nona.h.

6.445.2 Member Typedef Documentation

6.445.2.1 iterator

```
template<int RTYPE, bool NA>
typedef Rcpp::Vector<RTYPE>::const_iterator Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE >
>::iterator
```

Definition at line 49 of file nona.h.

6.445.2.2 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector<
RTYPE > >::STORAGE
```

Definition at line 48 of file nona.h.

6.445.2.3 SUGAR_TYPE

```
template<int RTYPE, bool NA>
typedef Rcpp::VectorBase<RTYPE,NA, Rcpp::Vector<RTYPE> > Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector<
RTYPE > >::SUGAR_TYPE
```

Definition at line 47 of file nona.h.

6.445.3 Constructor & Destructor Documentation

6.445.3.1 Nona()

```
template<int RTYPE, bool NA>
Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::Nona (
             const SUGAR_TYPE & expr )  [inline]
```

Definition at line 51 of file nona.h.

6.445.4 Member Function Documentation

6.445.4.1 operator[]()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 54 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

6.445.4.2 size()

```
template<int RTYPE, bool NA>
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::size ( ) const  [inline]
```

Definition at line 53 of file nona.h.

6.445.5 Member Data Documentation

6.445.5.1 data

```
template<int RTYPE, bool NA>
iterator Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::data  [private]
```

Definition at line 57 of file nona.h.

6.445.5.2 n

```
template<int RTYPE, bool NA>
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::n  [private]
```

Definition at line 58 of file nona.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

6.446 Rcpp::sugar::NonaPrimitive< T > Class Template Reference

```
#include <nona.h>
```

Public Member Functions

- NonaPrimitive (T t)
- operator T ()

Private Attributes

- T x

6.446.1 Detailed Description

template<typename T>
class Rcpp::sugar::NonaPrimitive< T >

Definition at line 62 of file nona.h.

6.446.2 Constructor & Destructor Documentation

6.446.2.1 NonaPrimitive()

```
template<typename T >
Rcpp::sugar::NonaPrimitive< T >::NonaPrimitive (
              T t ) [inline]
```

Definition at line 64 of file nona.h.

6.446.3 Member Function Documentation

6.446.3.1 operator T()

```
template<typename T >
Rcpp::sugar::NonaPrimitive< T >::operator T ( ) [inline]
```

Definition at line 65 of file nona.h.

References Rcpp::sugar::NonaPrimitive< T >::x.

6.446.4 Member Data Documentation

6.446.4.1 x

```
template<typename T >
T Rcpp::sugar::NonaPrimitive< T >::x  [private]
```

Definition at line 68 of file nona.h.

Referenced by Rcpp::sugar::NonaPrimitive< T >::operator T().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

6.447 Rcpp::NoProtectStorage< CLASS > Class Template Reference

```
#include <NoProtectStorage.h>
```

Public Member Functions

- NoProtectStorage ()
- ∼NoProtectStorage ()
- void set__ (SEXP x)
- SEXP get__ () const
- SEXP invalidate__ ()
- CLASS & copy__ (const CLASS &other)
- bool inherits (const char ∗clazz) const
- operator SEXP () const

Private Attributes

- SEXP data

6.447.1 Detailed Description

template<typename CLASS>
class Rcpp::NoProtectStorage< CLASS >

Definition at line 7 of file NoProtectStorage.h.

6.447.2 Constructor & Destructor Documentation

6.447.2.1 NoProtectStorage()

```
template<typename CLASS >
Rcpp::NoProtectStorage< CLASS >::NoProtectStorage ( )  [inline]
```

Definition at line 10 of file NoProtectStorage.h.

6.447.2.2 ∼NoProtectStorage()

```
template<typename CLASS >
Rcpp::NoProtectStorage< CLASS >::∼NoProtectStorage ( )  [inline]
```

Definition at line 12 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.447.3 Member Function Documentation

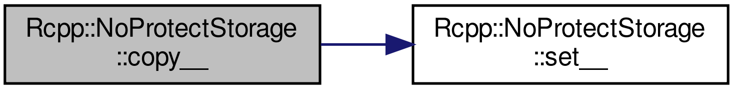
6.447.3.1 copy__()

```
template<typename CLASS >
CLASS& Rcpp::NoProtectStorage< CLASS >::copy__ (
            const CLASS & other )  [inline]
```

Definition at line 33 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::set__().

Here is the call graph for this function:



6.447.3.2 get__()

```
template<typename CLASS >
SEXP Rcpp::NoProtectStorage< CLASS >::get__ ( ) const  [inline]
```

Definition at line 24 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.447.3.3 inherits()

```
template<typename CLASS >
bool Rcpp::NoProtectStorage< CLASS >::inherits (
             const char * clazz ) const  [inline]
```

Definition at line 40 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.447.3.4 invalidate__()

```
template<typename CLASS >
SEXP Rcpp::NoProtectStorage< CLASS >::invalidate__ ( )  [inline]
```

Definition at line 28 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.447.3.5 operator SEXP()

```
template<typename CLASS >
Rcpp::NoProtectStorage< CLASS >::operator SEXP ( ) const  [inline]
```

Definition at line 44 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

6.447.3.6 set__()

```
template<typename CLASS >
void Rcpp::NoProtectStorage< CLASS >::set__ (
             SEXP x )  [inline]
```

Definition at line 16 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

Referenced by Rcpp::NoProtectStorage< CLASS >::copy__().

6.447.4 Member Data Documentation

6.447.4.1 data

```
template<typename CLASS >
SEXP Rcpp::NoProtectStorage< CLASS >::data  [private]
```

Definition at line 49 of file NoProtectStorage.h.

Referenced by Rcpp::NoProtectStorage< CLASS >::get__(), Rcpp::NoProtectStorage< CLASS >::inherits(), Rcpp↩ ::NoProtectStorage< CLASS >::invalidate__(), Rcpp::NoProtectStorage< CLASS >::operator SEXP(), Rcpp::No↩ ProtectStorage< CLASS >::set__(), and Rcpp::NoProtectStorage< CLASS >::∼NoProtectStorage().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/storage/NoProtectStorage.h

6.448 Rcpp::traits::normal_wrap_tag Struct Reference

```
#include <module_wrap_traits.h>
```

6.448.1 Detailed Description

Definition at line 29 of file module_wrap_traits.h.

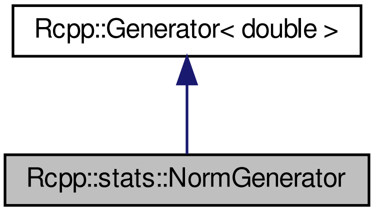
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

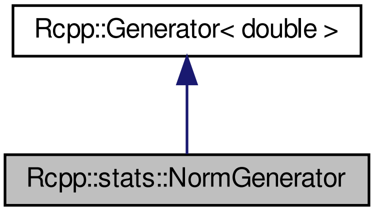
6.449 Rcpp::stats::NormGenerator Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator:



Collaboration diagram for Rcpp::stats::NormGenerator:



Public Member Functions

- NormGenerator (double mean_=0.0, double sd_=1.0)
- double operator() () const

Private Attributes

- double mean
- double sd

Additional Inherited Members

6.449.1 Detailed Description

Definition at line 28 of file rnorm.h.

6.449.2 Constructor & Destructor Documentation

6.449.2.1 NormGenerator()

```
Rcpp::stats::NormGenerator::NormGenerator (
            double mean_ = 0.0,
            double sd_ = 1.0 )  [inline]
```

Definition at line 31 of file rnorm.h.

6.449.3 Member Function Documentation

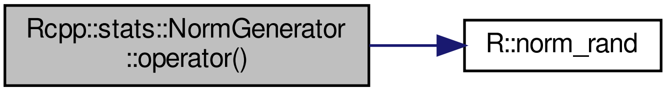
6.449.3.1 operator()()

```
double Rcpp::stats::NormGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rnorm.h.

References mean, R::norm_rand(), and sd.

Here is the call graph for this function:



6.449.4 Member Data Documentation

6.449.4.1 mean

```
double Rcpp::stats::NormGenerator::mean  [private]
```

Definition at line 39 of file rnorm.h.

Referenced by operator()().

6.449.4.2 sd

```
double Rcpp::stats::NormGenerator::sd  [private]
```

Definition at line 40 of file rnorm.h.

Referenced by operator()().

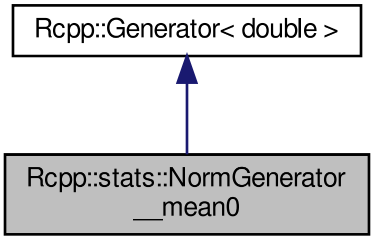
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

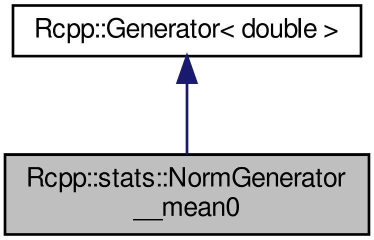
6.450 Rcpp::stats::NormGenerator__mean0 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator__mean0:



Collaboration diagram for Rcpp::stats::NormGenerator__mean0:



Public Member Functions

- NormGenerator__mean0 (double sd_=1.0)
- double operator() () const

Private Attributes

- double sd

Additional Inherited Members

6.450.1 Detailed Description

Definition at line 56 of file rnorm.h.

6.450.2 Constructor & Destructor Documentation

6.450.2.1 NormGenerator__mean0()

```
Rcpp::stats::NormGenerator__mean0::NormGenerator__mean0 (
            double sd_ = 1.0 )  [inline]
```

Definition at line 59 of file rnorm.h.

6.450.3 Member Function Documentation

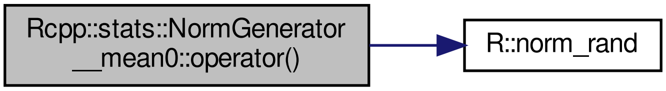
6.450.3.1 operator()()

```
double Rcpp::stats::NormGenerator__mean0::operator() ( ) const  [inline]
```

Definition at line 61 of file rnorm.h.

References R::norm_rand(), and sd.

Here is the call graph for this function:



6.450.4 Member Data Documentation

6.450.4.1 sd

```
double Rcpp::stats::NormGenerator__mean0::sd  [private]
```

Definition at line 66 of file rnorm.h.

Referenced by operator()().

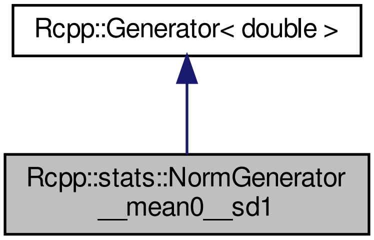
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

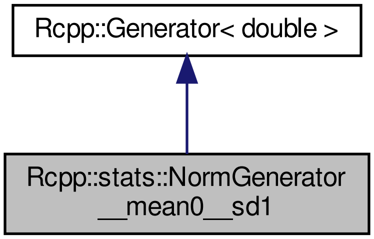
6.451 Rcpp::stats::NormGenerator__mean0__sd1 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator__mean0__sd1:


Rcpp::Generator< double >
Rcpp::stats::NormGenerator __mean0__sd1

Collaboration diagram for Rcpp::stats::NormGenerator__mean0__sd1:


Rcpp::Generator< double >
Rcpp::stats::NormGenerator __mean0__sd1

Public Member Functions

- NormGenerator__mean0__sd1 ()
- double operator() () const

Additional Inherited Members

6.451.1 Detailed Description

Definition at line 69 of file rnorm.h.

6.451.2 Constructor & Destructor Documentation

6.451.2.1 NormGenerator__mean0__sd1()

```
Rcpp::stats::NormGenerator__mean0__sd1::NormGenerator__mean0__sd1 ( )  [inline]
```

Definition at line 72 of file rnorm.h.

6.451.3 Member Function Documentation

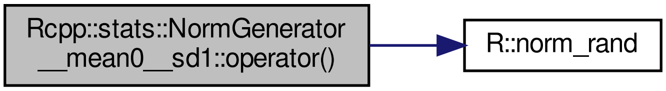
6.451.3.1 operator()()

```
double Rcpp::stats::NormGenerator__mean0__sd1::operator() ( ) const  [inline]
```

Definition at line 74 of file rnorm.h.

References R::norm_rand().

Here is the call graph for this function:



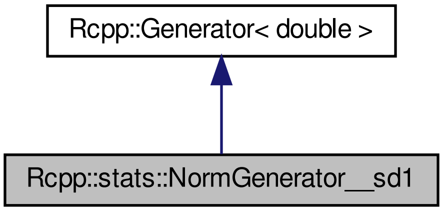
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

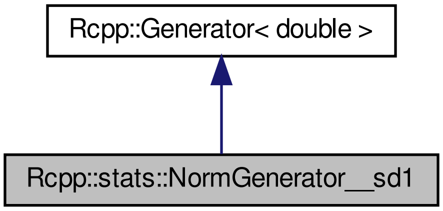
6.452 Rcpp::stats::NormGenerator__sd1 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator__sd1:



Collaboration diagram for Rcpp::stats::NormGenerator__sd1:



Public Member Functions

- NormGenerator__sd1 (double mean_=0.0)
- double operator() () const

Private Attributes

- double mean

Additional Inherited Members

6.452.1 Detailed Description

Definition at line 43 of file rnorm.h.

6.452.2 Constructor & Destructor Documentation

6.452.2.1 NormGenerator__sd1()

```
Rcpp::stats::NormGenerator__sd1::NormGenerator__sd1 (
             double mean_ = 0.0 )  [inline]
```

Definition at line 46 of file rnorm.h.

6.452.3 Member Function Documentation

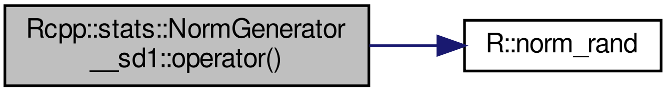
6.452.3.1 operator()()

```
double Rcpp::stats::NormGenerator__sd1::operator() ( ) const  [inline]
```

Definition at line 48 of file rnorm.h.

References mean, and R::norm_rand().

Here is the call graph for this function:



6.452.4 Member Data Documentation

6.452.4.1 mean

```
double Rcpp::stats::NormGenerator__sd1::mean [private]
```

Definition at line 53 of file rnorm.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rnorm.h

6.453 Rcpp::sugar::not_< RTYPE, NA > Class Template Reference

```
#include <not.h>
```

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- int apply (STORAGE x) const

6.453.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::not_< RTYPE, NA >

Definition at line 29 of file not.h.

6.453.2 Member Typedef Documentation

6.453.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::not_< RTYPE, NA >::STORAGE
```

Definition at line 31 of file not.h.

6.453.3 Member Function Documentation

6.453.3.1 apply()

```
template<int RTYPE, bool NA>
int Rcpp::sugar::not_< RTYPE, NA >::apply (
             STORAGE x ) const  [inline]
```

Definition at line 32 of file not.h.

Referenced by Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.454 Rcpp::sugar::not_< CPLXSXP, false > Class Reference

```
#include <not.h>
```

Public Member Functions

- int apply (Rcomplex x) const

6.454.1 Detailed Description

Definition at line 66 of file not.h.

6.454.2 Member Function Documentation

6.454.2.1 apply()

```
int Rcpp::sugar::not_< 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

6.455 Rcpp::sugar::not_< CPLXSXP, NA > Class Template Reference

```
#include <not.h>
```

Public Member Functions

- int apply (Rcomplex x) const

6.455.1 Detailed Description

template<bool NA>
class Rcpp::sugar::not_< CPLXSXP, NA >

Definition at line 59 of file not.h.

6.455.2 Member Function Documentation

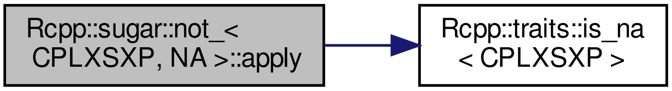
6.455.2.1 apply()

```
template<bool NA>
int Rcpp::sugar::not_< CPLXSXP, NA >::apply (
            Rcomplex x ) const  [inline]
```

Definition at line 61 of file not.h.

References Rcpp::traits::is_na< CPLXSXP >().

Here is the call graph for this function:


Rcpp::sugar::not_< CPLXSXP, NA >::apply
Rcpp::traits::is_na < CPLXSXP >

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.456 Rcpp::sugar::not_< REALSXP, false > Class Reference

```
#include <not.h>
```

Public Member Functions

- int apply (double x) const

6.456.1 Detailed Description

Definition at line 52 of file not.h.

6.456.2 Member Function Documentation

6.456.2.1 apply()

```
int Rcpp::sugar::not_< REALSXP, false >::apply (
            double x ) const  [inline]
```

Definition at line 54 of file not.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.457 Rcpp::sugar::not_< REALSXP, NA > Class Template Reference

```
#include <not.h>
```

Public Member Functions

- int apply (double x) const

6.457.1 Detailed Description

template<bool NA>
class Rcpp::sugar::not_< REALSXP, NA >

Definition at line 45 of file not.h.

6.457.2 Member Function Documentation

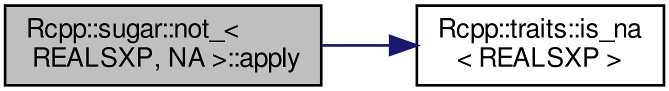
6.457.2.1 apply()

```
template<bool NA>
int Rcpp::sugar::not_< REALSXP, NA >::apply (
            double x ) const  [inline]
```

Definition at line 47 of file not.h.

References Rcpp::traits::is_na< REALSXP >().

Here is the call graph for this function:


Rcpp::sugar::not_< REALSXP, NA >::apply
Rcpp::traits::is_na < REALSXP >

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.458 Rcpp::sugar::not_< RTYPE, false > Class Template Reference

```
#include <not.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- int apply (STORAGE x) const

6.458.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::not_< RTYPE, false >

Definition at line 37 of file not.h.

6.458.2 Member Typedef Documentation

6.458.2.1 STORAGE

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::not_< RTYPE, false >::STORAGE
```

Definition at line 39 of file not.h.

6.458.3 Member Function Documentation

6.458.3.1 apply()

```
template<int RTYPE>
int Rcpp::sugar::not_< RTYPE, false >::apply (
             STORAGE x ) const  [inline]
```

Definition at line 40 of file not.h.

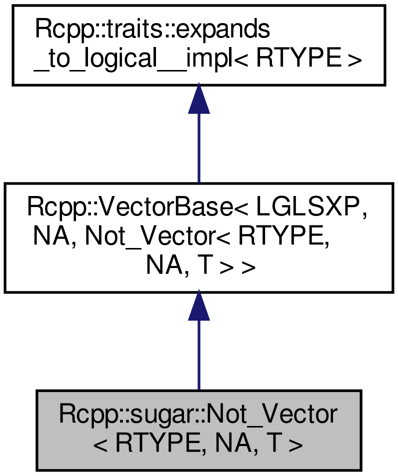
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

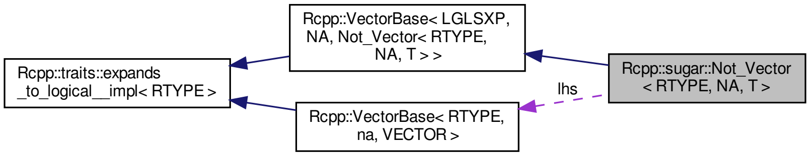
6.459 Rcpp::sugar::Not_Vector< RTYPE, NA, T > Class Template Reference

```
#include <not.h>
```

Inheritance diagram for Rcpp::sugar::Not_Vector< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE, NA, T > >
Rcpp::sugar::Not_Vector < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Not_Vector< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
Rcpp::sugar::Not_Vector < RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef not_< RTYPE, NA > OPERATOR

Public Member Functions

- Not_Vector (const VEC_TYPE &lhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & lhs
- OPERATOR op

6.459.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Not_Vector< RTYPE, NA, T >

Definition at line 76 of file not.h.

6.459.2 Member Typedef Documentation

6.459.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >
typedef not_<RTYPE,NA> Rcpp::sugar::Not_Vector< RTYPE, NA, T >::OPERATOR
```

Definition at line 80 of file not.h.

6.459.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Not_Vector< RTYPE, NA, T >::STORAGE
```

Definition at line 79 of file not.h.

6.459.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Not_Vector< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 78 of file not.h.

6.459.3 Constructor & Destructor Documentation

6.459.3.1 Not_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Not_Vector< RTYPE, NA, T >::Not_Vector (
             const VEC_TYPE & lhs_ )  [inline]
```

Definition at line 82 of file not.h.

6.459.4 Member Function Documentation

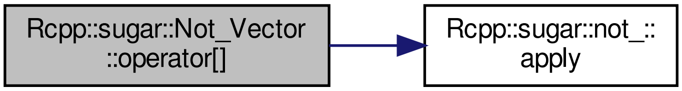
6.459.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 85 of file not.h.

References Rcpp::sugar::not_< RTYPE, NA >::apply(), Rcpp::sugar::Not_Vector< RTYPE, NA, T >::lhs, and Rcpp↩ ::sugar::Not_Vector< RTYPE, NA, T >::op.

Here is the call graph for this function:


Rcpp::sugar::Not_Vector ::operator[]
Rcpp::sugar::not_:: apply

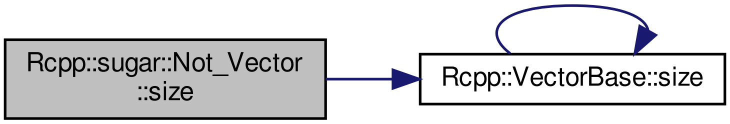
6.459.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Not_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 89 of file not.h.

References Rcpp::sugar::Not_Vector< RTYPE, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.459.5 Member Data Documentation

6.459.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Not_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 92 of file not.h.

Referenced by Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Not_Vector< RTYPE, NA, T >::size().

6.459.5.2 op

```
template<int RTYPE, bool NA, typename T >
OPERATOR Rcpp::sugar::Not_Vector< RTYPE, NA, T >::op [private]
```

Definition at line 93 of file not.h.

Referenced by Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

6.460 Rcpp::Nullable< T > Class Template Reference

```
#include <Nullable.h>
```

Public Member Functions

- Nullable ()
- Nullable (const T &t)
- Nullable (SEXP t)
- Nullable & operator= (SEXP sexp)
- operator SEXP () const
- SEXP get () const
- bool isUsable () const
- bool isNull () const
- bool isNotNull () const
- bool isSet (void) const
- T as ()
- T clone () const

Private Member Functions

- void checkIfSet (void) const

Private Attributes

- SEXP m_sexp
- bool m_set

6.460.1 Detailed Description

template<class T>
class Rcpp::Nullable< T >

Definition at line 33 of file Nullable.h.

6.460.2 Constructor & Destructor Documentation

6.460.2.1 Nullable() [1/3]

```
template<class T >
Rcpp::Nullable< T >::Nullable ( )  [inline]
```

Empty no-argument constructor of a Nullable object

Assigns (R's) NULL value, and sets validator to FALSE

Definition at line 41 of file Nullable.h.

6.460.2.2 Nullable() [2/3]

```
template<class T >
Rcpp::Nullable< T >::Nullable (
             const T & t )  [inline]
```

Template constructor of a Nullable object

Assigns object, and set validator to TRUE

Definition at line 49 of file Nullable.h.

6.460.2.3 Nullable() [3/3]

```
template<class T >
Rcpp::Nullable< T >::Nullable (
             SEXP t )  [inline]
```

Standard constructor of a Nullable object

Parameters

SEXP	is stored

Definition at line 56 of file Nullable.h.

References Rcpp::Nullable< T >::m_set, and Rcpp::Nullable< T >::m_sexp.

6.460.3 Member Function Documentation

6.460.3.1 as()

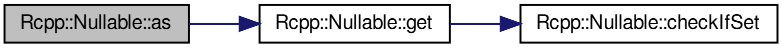
```
template<class T >
T Rcpp::Nullable< T >::as ( )  [inline]
```

Returns m_sexp as a T

Definition at line 129 of file Nullable.h.

References Rcpp::Nullable< T >::get().

Here is the call graph for this function:



6.460.3.2 checkIfSet()

```
template<class T >
void Rcpp::Nullable< T >::checkIfSet (
             void  ) const  [inline], [private]
```

Definition at line 140 of file Nullable.h.

References Rcpp::Nullable< T >::m_set.

Referenced by Rcpp::Nullable< T >::get(), Rcpp::Nullable< T >::isNull(), and Rcpp::Nullable< T >::operator SEXP().

6.460.3.3 clone()

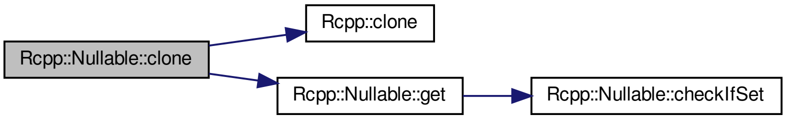
```
template<class T >
T Rcpp::Nullable< T >::clone ( ) const  [inline]
```

Return a clone of m_sexp as a T

Definition at line 134 of file Nullable.h.

References Rcpp::clone(), and Rcpp::Nullable< T >::get().

Here is the call graph for this function:



6.460.3.4 get()

```
template<class T >
SEXP Rcpp::Nullable< T >::get ( ) const  [inline]
```

get() accessor for object

Exceptions

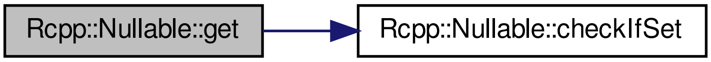
'not	initialized' if object has not been set

Definition at line 89 of file Nullable.h.

References Rcpp::Nullable< T >::checkIfSet(), and Rcpp::Nullable< T >::m_sexp.

Referenced by Rcpp::Nullable< T >::as(), and Rcpp::Nullable< T >::clone().

Here is the call graph for this function:



6.460.3.5 isNotNull()

```
template<class T >
bool Rcpp::Nullable< T >::isNotNull ( ) const  [inline]
```

Boolean test for not NULL

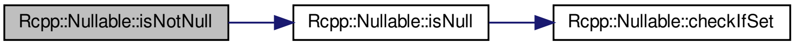
Exceptions

'not	initialized' if object has not been set

Definition at line 116 of file Nullable.h.

References Rcpp::Nullable< T >::isNull().

Here is the call graph for this function:



6.460.3.6 isNull()

```
template<class T >
bool Rcpp::Nullable< T >::isNull ( ) const  [inline]
```

Boolean test for NULL

Exceptions

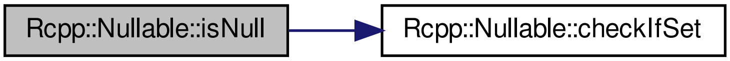
'not	initialized' if object has not been set

Definition at line 106 of file Nullable.h.

References Rcpp::Nullable< T >::checkIfSet(), and Rcpp::Nullable< T >::m_sexp.

Referenced by Rcpp::Nullable< T >::isNotNull().

Here is the call graph for this function:



6.460.3.7 isSet()

```
template<class T >
bool Rcpp::Nullable< T >::isSet (
             void  ) const  [inline]
```

Test function to check if object has been initialized

Definition at line 124 of file Nullable.h.

References Rcpp::Nullable< T >::m_set.

6.460.3.8 isUsable()

```
template<class T >
bool Rcpp::Nullable< T >::isUsable ( ) const  [inline]
```

Boolean test for usability as a T

Definition at line 97 of file Nullable.h.

References Rcpp::Nullable< T >::m_set, and Rcpp::Nullable< T >::m_sexp.

6.460.3.9 operator SEXP()

```
template<class T >
Rcpp::Nullable< T >::operator SEXP ( ) const  [inline]
```

operator SEXP() to return nullable object

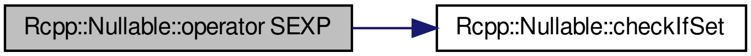
Exceptions

'not	initialized' if object has not been set

Definition at line 79 of file Nullable.h.

References Rcpp::Nullable< T >::checkIfSet(), and Rcpp::Nullable< T >::m_sexp.

Here is the call graph for this function:



6.460.3.10 operator=()

```
template<class T >
Nullable& Rcpp::Nullable< T >::operator= (
             SEXP sexp )  [inline]
```

Copy constructor for Nullable object

Parameters

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.

6.460.4 Member Data Documentation

6.460.4.1 m_set

```
template<class T >
bool Rcpp::Nullable< T >::m_set  [private]
```

Definition at line 138 of file Nullable.h.

Referenced by Rcpp::Nullable< T >::checkIfSet(), Rcpp::Nullable< T >::isSet(), Rcpp::Nullable< T >::isUsable(), Rcpp::Nullable< T >::Nullable(), and Rcpp::Nullable< T >::operator=().

6.460.4.2 m_sexp

```
template<class T >
SEXP Rcpp::Nullable< T >::m_sexp  [private]
```

Definition at line 137 of file Nullable.h.

Referenced by Rcpp::Nullable< T >::get(), Rcpp::Nullable< T >::isNull(), Rcpp::Nullable< T >::isUsable(), Rcpp::Nullable< T >::Nullable(), Rcpp::Nullable< T >::operator SEXP(), and Rcpp::Nullable< T >::operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Nullable.h

6.461 Rcpp::traits::num2type< N > Struct Template Reference

```
#include <num2type.h>
```

6.461.1 Detailed Description

template<std::size_t N>
struct Rcpp::traits::num2type< N >

Definition at line 29 of file num2type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/num2type.h

6.462 Rcpp::object< T > Class Template Reference

```
#include <Module.h>
```

Public Types

- typedef T object_type

Public Member Functions

- object (T *ptr_)
- operator T* ()
- T * operator-> ()
- T & operator& ()

Public Attributes

- T ∗ ptr

6.462.1 Detailed Description

template<typename T>
class Rcpp::object< T >

Definition at line 65 of file Module.h.

6.462.2 Member Typedef Documentation

6.462.2.1 object_type

```
template<typename T >
typedef T Rcpp::object< T >::object_type
```

Definition at line 67 of file Module.h.

6.462.3 Constructor & Destructor Documentation

6.462.3.1 object()

```
template<typename T >
Rcpp::object< T >::object (
              T ∗ ptr_ )  [inline]
```

Definition at line 68 of file Module.h.

6.462.4 Member Function Documentation

6.462.4.1 operator T∗()

```
template<typename T >
Rcpp::object< T >::operator T* ( )  [inline]
```

Definition at line 69 of file Module.h.

References Rcpp::object< T >::ptr.

6.462.4.2 operator&()

```
template<typename T >
T& Rcpp::object< T >::operator& ( )  [inline]
```

Definition at line 71 of file Module.h.

References Rcpp::object< T >::ptr.

6.462.4.3 operator->()

```
template<typename T >
T* Rcpp::object< T >::operator-> ( )  [inline]
```

Definition at line 70 of file Module.h.

References Rcpp::object< T >::ptr.

6.462.5 Member Data Documentation

6.462.5.1 ptr

```
template<typename T >
T* Rcpp::object< T >::ptr
```

Definition at line 72 of file Module.h.

Referenced by Rcpp::object< T >::operator T∗(), Rcpp::object< T >::operator&(), and Rcpp::object< T >::operator->().

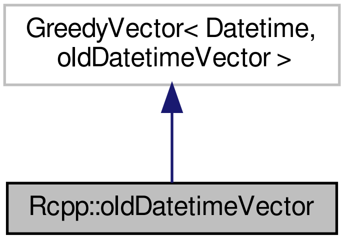
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

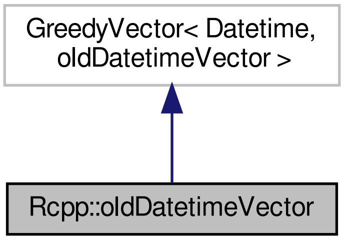
6.463 Rcpp::oldDatetimeVector Class Reference

```
#include <oldDatetimeVector.h>
```

Inheritance diagram for Rcpp::oldDatetimeVector:



Collaboration diagram for Rcpp::oldDatetimeVector:



Public Member Functions

- oldDatetimeVector (SEXP vec)
- oldDatetimeVector (int n)
- std::vector< Datetime > getDatetimes () const

6.463.1 Detailed Description

Definition at line 31 of file oldDatetimeVector.h.

6.463.2 Constructor & Destructor Documentation

6.463.2.1 oldDatetimeVector() [1/2]

```
Rcpp::oldDatetimeVector::oldDatetimeVector (
             SEXP vec )  [inline]
```

Definition at line 33 of file oldDatetimeVector.h.

6.463.2.2 oldDatetimeVector() [2/2]

```
Rcpp::oldDatetimeVector::oldDatetimeVector (
             int n )  [inline]
```

Definition at line 34 of file oldDatetimeVector.h.

6.463.3 Member Function Documentation

6.463.3.1 getDatetimes()

```
std::vector<Datetime> Rcpp::oldDatetimeVector::getDatetimes ( ) const  [inline]
```

Definition at line 36 of file oldDatetimeVector.h.

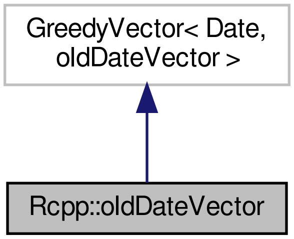
The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/oldDatetimeVector.h

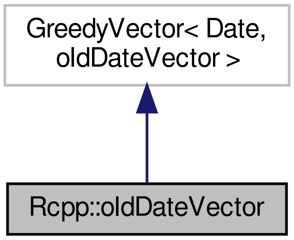
6.464 Rcpp::oldDateVector Class Reference

```
#include <oldDateVector.h>
```

Inheritance diagram for Rcpp::oldDateVector:



Collaboration diagram for Rcpp::oldDateVector:



Public Member Functions

- oldDateVector (SEXP vec)
- oldDateVector (int n)
- std::vector< Date > getDates () const

6.464.1 Detailed Description

Definition at line 30 of file oldDateVector.h.

6.464.2 Constructor & Destructor Documentation

6.464.2.1 oldDateVector() [1/2]

```
Rcpp::oldDateVector::oldDateVector (
             SEXP vec )  [inline]
```

Definition at line 32 of file oldDateVector.h.

6.464.2.2 oldDateVector() [2/2]

```
Rcpp::oldDateVector::oldDateVector (
             int n )  [inline]
```

Definition at line 33 of file oldDateVector.h.

6.464.3 Member Function Documentation

6.464.3.1 getDates()

```
std::vector<Date> Rcpp::oldDateVector::getDates ( ) const  [inline]
```

Definition at line 35 of file oldDateVector.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/date_datetime/oldDateVector.h

6.465 Rcpp::traits::one_type< T > Class Template Reference

```
#include <one_type.h>
```

Public Member Functions

- operator T () const

Private Member Functions

- Rcomplex op (true_type) const
- T op (false_type) const

6.465.1 Detailed Description

template<typename T>
class Rcpp::traits::one_type< T >

Definition at line 35 of file one_type.h.

6.465.2 Member Function Documentation

6.465.2.1 op() [1/2]

```
template<typename T >
T Rcpp::traits::one_type< T >::op (
             false_type   ) const  [inline], [private]
```

Definition at line 44 of file one_type.h.

6.465.2.2 op() [2/2]

```
template<typename T >
Rcomplex Rcpp::traits::one_type< T >::op (
             true_type   ) const  [inline], [private]
```

Definition at line 37 of file one_type.h.

Referenced by Rcpp::traits::one_type< T >::operator T().

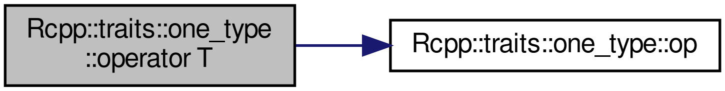
6.465.2.3 operator T()

```
template<typename T >
Rcpp::traits::one_type< T >::operator T ( ) const [inline]
```

Definition at line 49 of file one_type.h.

References Rcpp::traits::one_type< T >::op().

Here is the call graph for this function:


Rcpp::traits::one_type ::operator T
Rcpp::traits::one_type::op

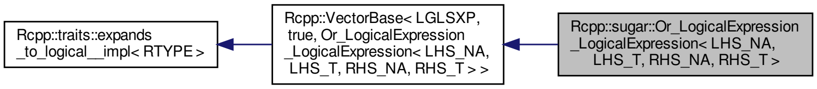
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/one_type.h

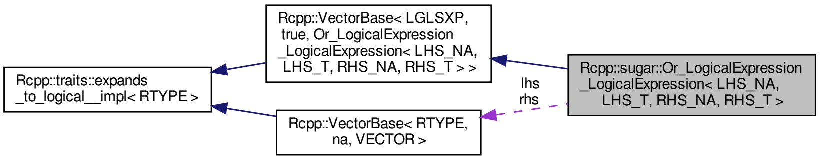
6.466 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
lhs
rhs
Rcpp::sugar::Or_LogicalExpression _LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > RHS_TYPE

Public Member Functions

- Or_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.466.1 Detailed Description

template< bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 202 of file or.h.

6.466.2 Member Typedef Documentation

6.466.2.1 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,LHS_NA,LHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 204 of file or.h.

6.466.2.2 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,RHS_NA,RHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 205 of file or.h.

6.466.3 Constructor & Destructor Documentation

6.466.3.1 Or_LogicalExpression_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::Or_LogicalExpression_Logical
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 207 of file or.h.

6.466.4 Member Function Documentation

6.466.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 209 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

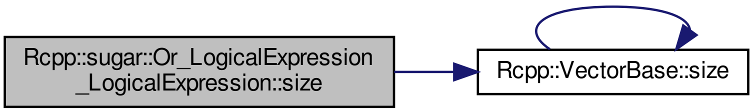
6.466.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::size ( ) const  [inline]
```

Definition at line 214 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.466.5 Member Data Documentation

6.466.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T
>::lhs  [private]
```

Definition at line 217 of file or.h.

Referenced by Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size().

6.466.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T
>::rhs  [private]
```

Definition at line 218 of file or.h.

Referenced by Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[ ](), and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[ ]().

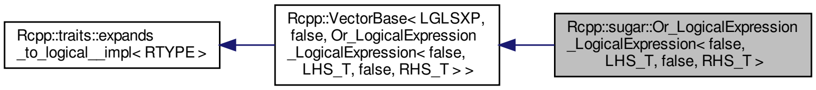
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

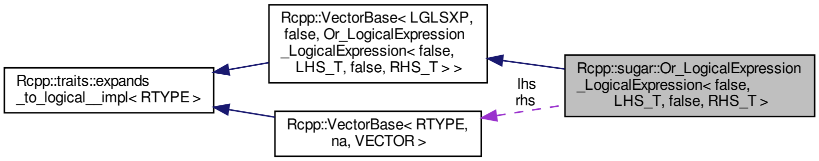
6.467 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, Or_LogicalExpression _LogicalExpression< false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, false, Or_LogicalExpression _LogicalExpression< false, LHS_T, false, RHS_T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
lhs
rhs
Rcpp::sugar::Or_LogicalExpression _LogicalExpression< false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, false, RHS_T > RHS_TYPE

Public Member Functions

- Or_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.467.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >

Definition at line 261 of file or.h.

6.467.2 Member Typedef Documentation

6.467.2.1 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,LHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 264 of file or.h.

6.467.2.2 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,RHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 265 of file or.h.

6.467.3 Constructor & Destructor Documentation

6.467.3.1 Or_LogicalExpression_LogicalExpression()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::Or_LogicalExpression_LogicalE
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 267 of file or.h.

6.467.4 Member Function Documentation

6.467.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 269 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

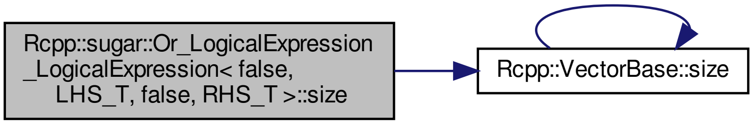
6.467.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size
( ) const  [inline]
```

Definition at line 273 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:


Rcpp::sugar::Or_LogicalExpression
_LogicalExpression< false,
LHS_T, false, RHS_T >::size
Rcpp::VectorBase::size

6.467.5 Member Data Documentation

6.467.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 276 of file or.h.

6.467.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 277 of file or.h.

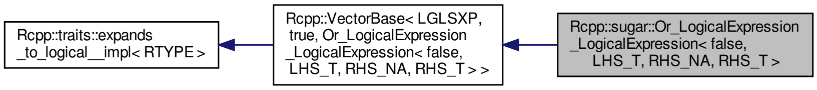
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

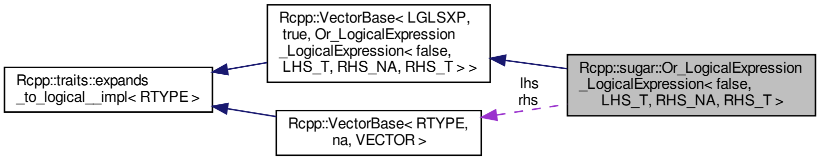
6.468 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
lhs
rhs
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > RHS_TYPE

Public Member Functions

- Or_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.468.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >

Definition at line 221 of file or.h.

6.468.2 Member Typedef Documentation

6.468.2.1 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,LHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 224 of file or.h.

6.468.2.2 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,RHS_NA,RHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 225 of file or.h.

6.468.3 Constructor & Destructor Documentation

6.468.3.1 Or_LogicalExpression_LogicalExpression()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::Or_LogicalExpression_LogicalE
(
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 227 of file or.h.

6.468.4 Member Function Documentation

6.468.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[]
(
             R_xlen_t i ) const  [inline]
```

Definition at line 229 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

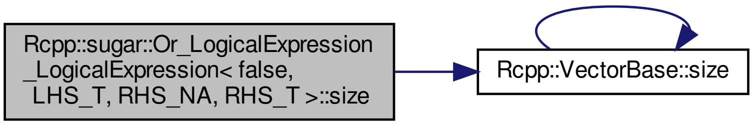
6.468.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size
( ) const  [inline]
```

Definition at line 234 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:


Rcpp::sugar::Or_LogicalExpression
_LogicalExpression< false,
LHS_T, RHS_NA, RHS_T >::size
Rcpp::VectorBase::size

6.468.5 Member Data Documentation

6.468.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::lhs [private]
```

Definition at line 237 of file or.h.

6.468.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::rhs [private]
```

Definition at line 238 of file or.h.

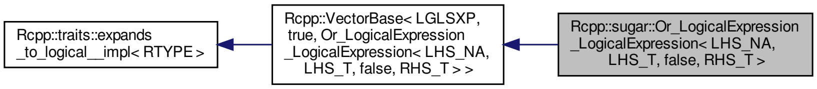
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

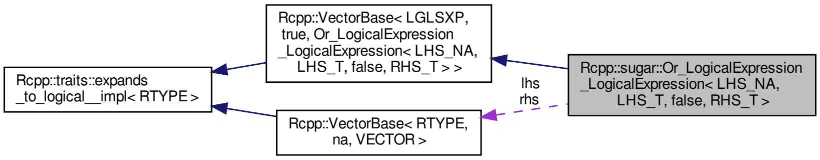
6.469 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
lhs
rhs
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< LGLSXP, false, RHS_T > RHS_TYPE

Public Member Functions

- Or_LogicalExpression_LogicalExpression (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

6.469.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >

Definition at line 241 of file or.h.

6.469.2 Member Typedef Documentation

6.469.2.1 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,LHS_NA,LHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 244 of file or.h.

6.469.2.2 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,RHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 245 of file or.h.

6.469.3 Constructor & Destructor Documentation

6.469.3.1 Or_LogicalExpression_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::Or_LogicalExpression_LogicalE
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 247 of file or.h.

6.469.4 Member Function Documentation

6.469.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 249 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

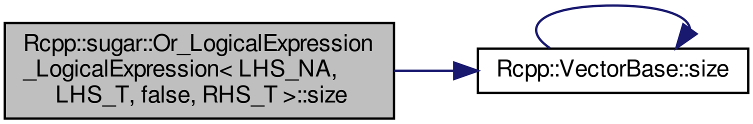
6.469.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size
( ) const  [inline]
```

Definition at line 254 of file or.h.

References Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:


Rcpp::sugar::Or_LogicalExpression
_LogicalExpression< LHS_NA,
LHS_T, false, RHS_T >::size
Rcpp::VectorBase::size

6.469.5 Member Data Documentation

6.469.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::lhs  [private]
```

Definition at line 257 of file or.h.

6.469.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::rhs  [private]
```

Definition at line 258 of file or.h.

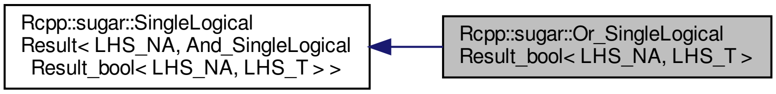
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

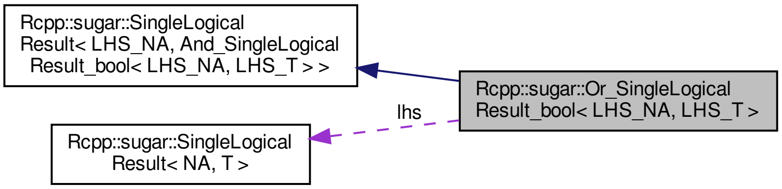
6.470 Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >:


Rcpp::sugar::SingleLogical
Result< LHS_NA, And_SingleLogical
Result_bool< LHS_NA, LHS_T > >
Rcpp::sugar::Or_SingleLogical
Result_bool< LHS_NA, LHS_T >

Collaboration diagram for Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, And_SingleLogical Result_bool< LHS_NA, LHS_T > >
Rcpp::sugar::Or_SingleLogical Result_bool< LHS_NA, LHS_T >
lhs
Rcpp::sugar::SingleLogical Result< NA, T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_bool< LHS_NA, LHS_T > > BASE

Public Member Functions

- Or_SingleLogicalResult_bool (const LHS_TYPE &lhs_, bool rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- bool rhs

Additional Inherited Members

6.470.1 Detailed Description

template<bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >

Definition at line 170 of file or.h.

6.470.2 Member Typedef Documentation

6.470.2.1 BASE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult< LHS_NA , Or_SingleLogicalResult_bool<LHS_NA,LHS_T> > Rcpp::sugar::Or_SingleLogicalR
LHS_NA, LHS_T >::BASE
```

Definition at line 181 of file or.h.

6.470.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS↩
_T >::LHS_TYPE
```

Definition at line 177 of file or.h.

6.470.3 Constructor & Destructor Documentation

6.470.3.1 Or_SingleLogicalResult_bool()

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::Or_SingleLogicalResult_bool (
             const LHS_TYPE & lhs_,
             bool rhs_ )  [inline]
```

Definition at line 183 of file or.h.

6.470.4 Member Function Documentation

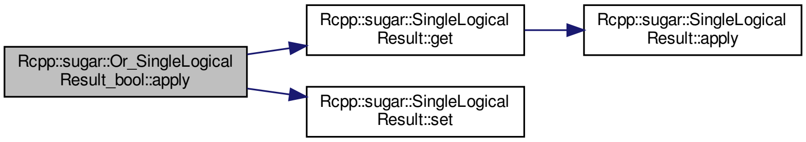
6.470.4.1 apply()

```
template<bool LHS_NA, typename LHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply ( )  [inline]
```

Definition at line 186 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs, and Rcpp::sugar::SingleLogical↩ Result< NA, T >::set().

Here is the call graph for this function:


Rcpp::sugar::Or_SingleLogical Result_bool::apply
Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::SingleLogical Result::set

6.470.5 Member Data Documentation

6.470.5.1 lhs

```
template<bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 195 of file or.h.

Referenced by Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

6.470.5.2 rhs

```
template<bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs  [private]
```

Definition at line 196 of file or.h.

Referenced by Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

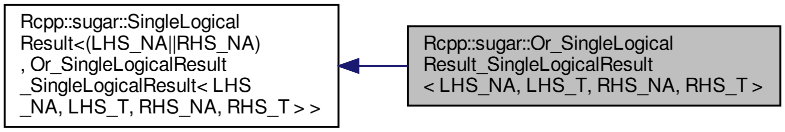
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

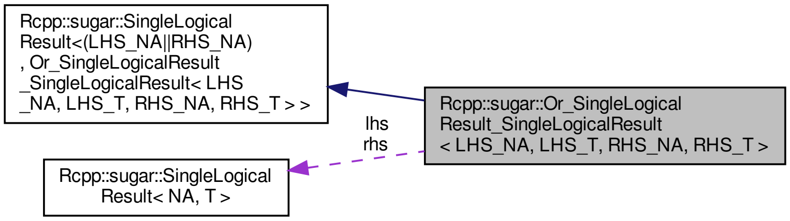
6.471 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, Or_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Or_SingleLogical
Result_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 >:


Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, Or_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T > >
lhs
rhs
Rcpp::sugar::Or_SingleLogical
Result_SingleLogicalResult
< LHS_NA, LHS_T, RHS_NA, RHS_T >
Rcpp::sugar::SingleLogical
Result< NA, T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< RHS_NA, RHS_T > RHS_TYPE
- typedef SingleLogicalResult<(LHS_NA||RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > > BASE

Public Member Functions

- Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.471.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file or.h.

6.471.2 Member Typedef Documentation

6.471.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< (LHS_NA || RHS_NA) , Or_SingleLogicalResult_SingleLogicalResult<LHS↩
_NA,LHS_T,RHS_NA,RHS_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T,
RHS_NA, RHS_T >::BASE
```

Definition at line 41 of file or.h.

6.471.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 36 of file or.h.

6.471.2.3 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 37 of file or.h.

6.471.3 Constructor & Destructor Documentation

6.471.3.1 Or_SingleLogicalResult_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::Or_SingleLogicalResult_S
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 43 of file or.h.

6.471.4 Member Function Documentation

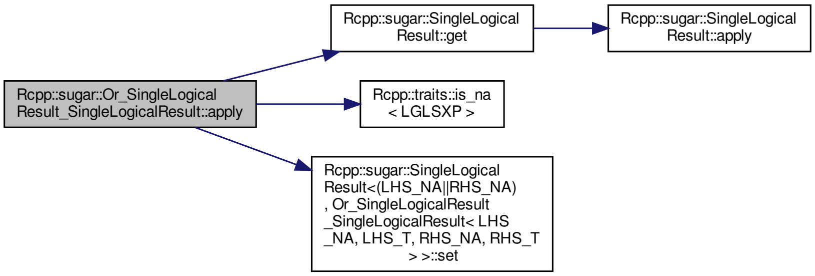
6.471.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 46 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::traits::is_na< LGLSXP >(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult::apply
Rcpp::traits::is_na < LGLSXP >
Rcpp::sugar::SingleLogical Result<(LHS_NA||RHS_NA) , Or_SingleLogicalResult _SingleLogicalResult< LHS _NA, LHS_T, RHS_NA, RHS_T > >::set

6.471.5 Member Data Documentation

6.471.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 58 of file or.h.

Referenced by Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >↩ ::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), and Rcpp↩ ::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply().

6.471.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 59 of file or.h.

Referenced by Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >↩ ::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), and Rcpp↩ ::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply().

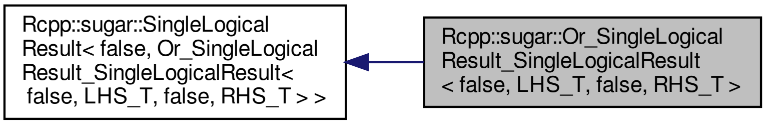
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

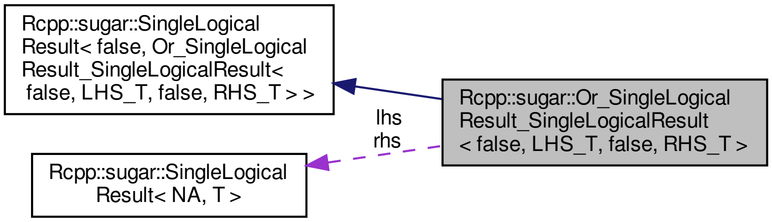
6.472 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical
Result< false, Or_SingleLogical
Result_SingleLogicalResult<
false, LHS_T, false, RHS_T > >
Rcpp::sugar::Or_SingleLogical
Result_SingleLogicalResult
< false, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical
Result< false, Or_SingleLogical
Result_SingleLogicalResult<
false, LHS_T, false, RHS_T > >
lhs
rhs
Rcpp::sugar::Or_SingleLogical
Result_SingleLogicalResult
< false, LHS_T, false, RHS_T >
Rcpp::sugar::SingleLogical
Result< NA, T >

Public Types

- typedef SingleLogicalResult< false, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< false, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< false, Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > > BASE

Public Member Functions

- Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.472.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >

Definition at line 136 of file or.h.

6.472.2 Member Typedef Documentation

6.472.2.1 BASE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< false, Or_SingleLogicalResult_SingleLogicalResult<false,LHS_T,false,RHS↩
_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >↩
::BASE
```

Definition at line 148 of file or.h.

6.472.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 143 of file or.h.

6.472.2.3 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 144 of file or.h.

6.472.3 Constructor & Destructor Documentation

6.472.3.1 Or_SingleLogicalResult_SingleLogicalResult()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::Or_SingleLogicalResult_Si
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 150 of file or.h.

6.472.4 Member Function Documentation

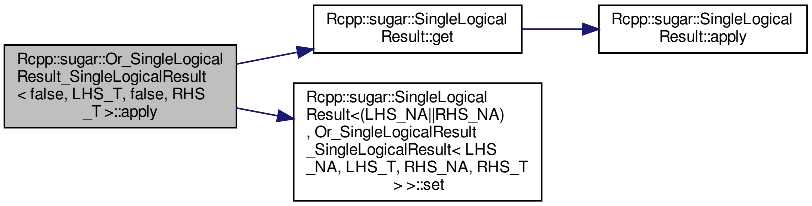
6.472.4.1 apply()

```
template<typename LHS_T , typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply
( )  [inline]
```

Definition at line 153 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::Or_SingleLogical
Result_SingleLogicalResult
< false, LHS_T, false, RHS
_T >::apply
Rcpp::sugar::SingleLogical
Result::get
Rcpp::sugar::SingleLogical
Result::apply
Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, Or_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T
> >::set

6.472.5 Member Data Documentation

6.472.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 163 of file or.h.

6.472.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 164 of file or.h.

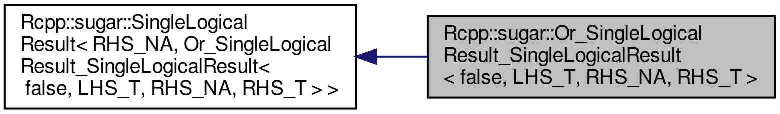
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

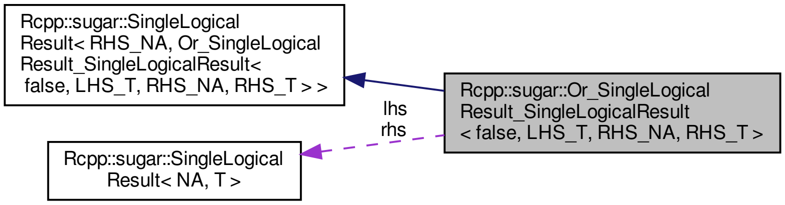
6.473 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::sugar::SingleLogical Result< RHS_NA, Or_SingleLogical Result_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult < false, LHS_T, RHS_NA, RHS_T >

Collaboration diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >:


Rcpp::sugar::SingleLogical Result< RHS_NA, Or_SingleLogical Result_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > >
lhs
rhs
Rcpp::sugar::SingleLogical Result< NA, T >
Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult < false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef SingleLogicalResult< false, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< RHS_NA, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< RHS_NA, Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > > BASE

Public Member Functions

- Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.473.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >

Definition at line 101 of file or.h.

6.473.2 Member Typedef Documentation

6.473.2.1 BASE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< RHS_NA, Or_SingleLogicalResult_SingleLogicalResult<false,LHS_T,RHS←
_NA,RHS_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS←
_T >::BASE
```

Definition at line 113 of file or.h.

6.473.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file or.h.

6.473.2.3 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file or.h.

6.473.3 Constructor & Destructor Documentation

6.473.3.1 Or_SingleLogicalResult_SingleLogicalResult()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::Or_SingleLogicalResult_S
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )   [inline]
```

Definition at line 115 of file or.h.

6.473.4 Member Function Documentation

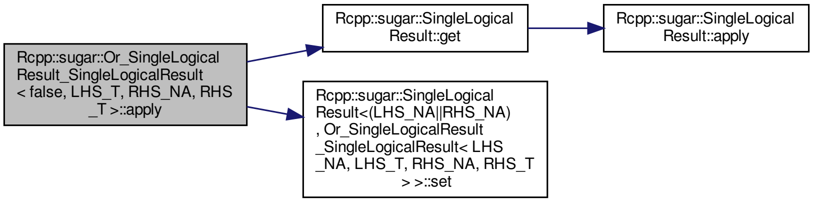
6.473.4.1 apply()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 118 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogical↩ Result< 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_Single↩ LogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult < false, LHS_T, RHS_NA, RHS _T >::apply
Rcpp::sugar::SingleLogical Result::get
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::SingleLogical Result<(LHS_NA||RHS_NA) , Or_SingleLogicalResult _SingleLogicalResult< LHS _NA, LHS_T, RHS_NA, RHS_T > >::set

6.473.5 Member Data Documentation

6.473.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::lhs  [private]
```

Definition at line 129 of file or.h.

6.473.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::rhs  [private]
```

Definition at line 130 of file or.h.

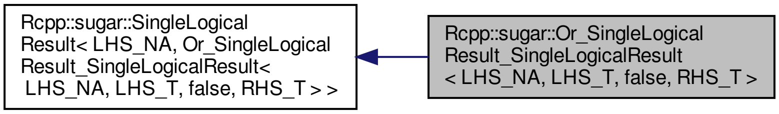
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

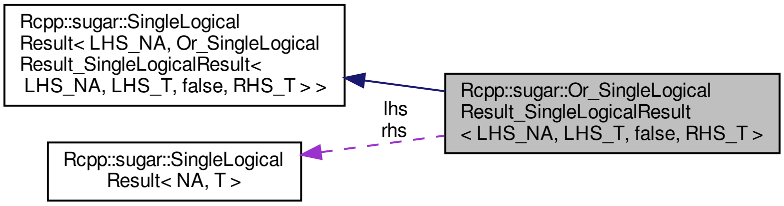
6.474 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_↩ T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, Or_SingleLogical Result_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult < LHS_NA, LHS_T, false, RHS_T >

Collaboration diagram for Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >:


Rcpp::sugar::SingleLogical Result< LHS_NA, Or_SingleLogical Result_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > >
lhs
rhs
Rcpp::sugar::Or_SingleLogical Result_SingleLogicalResult < LHS_NA, LHS_T, false, RHS_T >
Rcpp::sugar::SingleLogical Result< NA, T >

Public Types

- typedef SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE
- typedef SingleLogicalResult< false, RHS_T > RHS_TYPE
- typedef SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > > BASE

Public Member Functions

- Or_SingleLogicalResult_SingleLogicalResult (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- void apply ()

Private Attributes

- const LHS_TYPE & lhs
- const RHS_TYPE & rhs

Additional Inherited Members

6.474.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >

Definition at line 65 of file or.h.

6.474.2 Member Typedef Documentation

6.474.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_↩
T,false,RHS_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::BASE
```

Definition at line 77 of file or.h.

6.474.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 72 of file or.h.

6.474.2.3 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 73 of file or.h.

6.474.3 Constructor & Destructor Documentation

6.474.3.1 Or_SingleLogicalResult_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::Or_SingleLogicalResult_Si
(
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 79 of file or.h.

6.474.4 Member Function Documentation

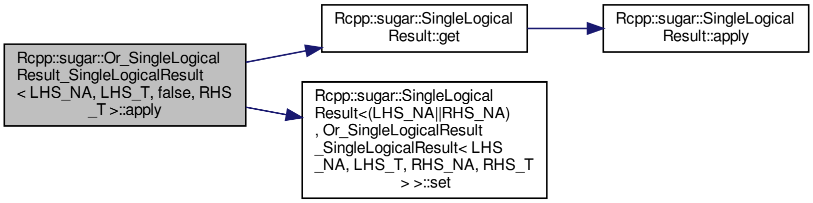
6.474.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >↩
::apply ( )  [inline]
```

Definition at line 82 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set().

Here is the call graph for this function:


Rcpp::sugar::Or_SingleLogical
Result_SingleLogicalResult
< LHS_NA, LHS_T, false, RHS
_T >::apply
Rcpp::sugar::SingleLogical
Result::get
Rcpp::sugar::SingleLogical
Result::apply
Rcpp::sugar::SingleLogical
Result<(LHS_NA||RHS_NA)
, Or_SingleLogicalResult
_SingleLogicalResult< LHS
_NA, LHS_T, RHS_NA, RHS_T
> >::set

6.474.5 Member Data Documentation

6.474.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 93 of file or.h.

6.474.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 94 of file or.h.

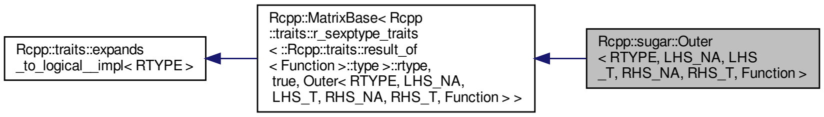
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

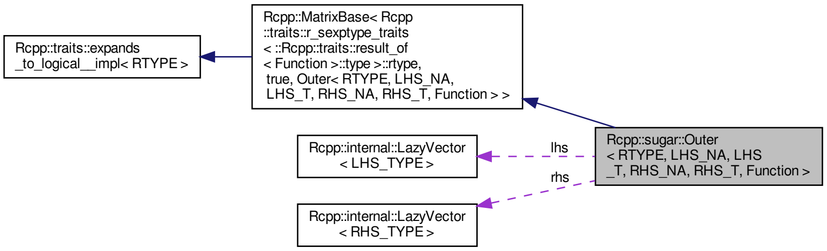
6.475 Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > Class Template Reference

```
#include <outer.h>
```

Inheritance diagram for Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::internal::LazyVector < LHS_TYPE >
lhs
rhs
Rcpp::sugar::Outer < RTYPE, LHS_NA, LHS _T, RHS_NA, RHS_T, Function >
Rcpp::internal::LazyVector < RHS_TYPE >

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type result_type
- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::internal::LazyVector< LHS_TYPE > LHS_LAZY
- typedef Rcpp::internal::LazyVector< RHS_TYPE > RHS_LAZY
- typedef Rcpp::traits::r_vector_element_converter< RESULT_R_TYPE >::type converter_type
- typedef Rcpp::traits::storage_type< RESULT_R_TYPE >::type STORAGE

Public Member Functions

- Outer (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_, Function fun_)
- STORAGE operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Static Public Attributes

- static const int RESULT_R_TYPE

Private Attributes

- LHS_LAZY lhs
- RHS_LAZY rhs
- Function fun
- int nr
- int nc

Additional Inherited Members

6.475.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T, typename Function>
class Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >

Definition at line 32 of file outer.h.

6.475.2 Member Typedef Documentation

6.475.2.1 converter_type

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::traits::r_vector_element_converter<RESULT_R_TYPE>::type Rcpp::sugar::Outer< RTYPE,
LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::converter_type
```

Definition at line 50 of file outer.h.

6.475.2.2 LHS_LAZY

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::internal::LazyVector<LHS_TYPE> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::LHS_LAZY
```

Definition at line 47 of file outer.h.

6.475.2.3 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::LHS_TYPE
```

Definition at line 44 of file outer.h.

6.475.2.4 result_type

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS←↩
_NA, RHS_T, Function >::result_type
```

Definition at line 40 of file outer.h.

6.475.2.5 RHS_LAZY

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::internal::LazyVector<RHS_TYPE> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::RHS_LAZY
```

Definition at line 48 of file outer.h.

6.475.2.6 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::RHS_TYPE
```

Definition at line 45 of file outer.h.

6.475.2.7 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T, Function >::STORAGE
```

Definition at line 51 of file outer.h.

6.475.3 Constructor & Destructor Documentation

6.475.3.1 Outer()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::Outer (
                const LHS_TYPE & lhs_,
                const RHS_TYPE & rhs_,
                Function fun_ )  [inline]
```

Definition at line 53 of file outer.h.

6.475.4 Member Function Documentation

6.475.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::ncol ( ) const  [inline]
```

Definition at line 62 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nc.

6.475.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nrow ( ) const  [inline]
```

Definition at line 61 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nr.

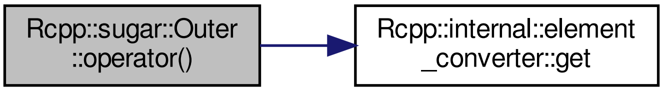
6.475.4.3 operator()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
STORAGE Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 56 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::fun, Rcpp::internal↩::element_converter< RTYPE >::get(), Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::lhs, and Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::rhs.

Here is the call graph for this function:



6.475.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
R_xlen_t Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::size ( ) const
[inline]
```

Definition at line 60 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nc, and Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nr.

6.475.5 Member Data Documentation

6.475.5.1 fun

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
Function Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::fun  [private]
```

Definition at line 69 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()().

6.475.5.2 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
LHS_LAZY Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::lhs  [private]
```

Definition at line 66 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()().

6.475.5.3 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nc  [private]
```

Definition at line 70 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::ncol(), and Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::size().

6.475.5.4 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nr  [private]
```

Definition at line 70 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nrow(), and Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::size().

6.475.5.5 RESULT_R_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
const int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::RESULT_R_TYPE
[static]
```

Initial value:

```
=
        Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 41 of file outer.h.

6.475.5.6 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
RHS_LAZY Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::rhs  [private]
```

Definition at line 67 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()().

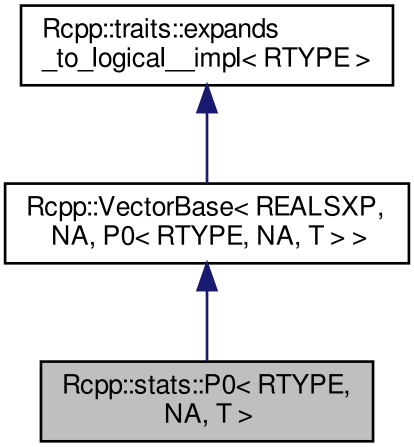
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/outer.h

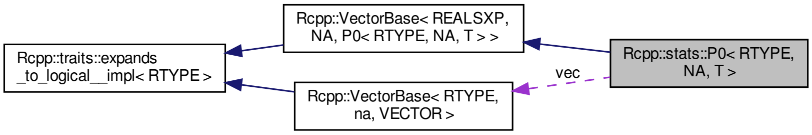
6.476 Rcpp::stats::P0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P0< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > >
Rcpp::stats::P0< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::P0< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::P0< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, int, int)

Public Member Functions

- P0 (FunPtr ptr_, const VEC_TYPE &vec_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- int lower
- int log

6.476.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P0< RTYPE, NA, T >

Definition at line 123 of file dpq.h.

6.476.2 Member Typedef Documentation

6.476.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::P0< RTYPE, NA, T >::FunPtr) (double, int, int)
```

Definition at line 126 of file dpq.h.

6.476.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P0< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 125 of file dpq.h.

6.476.3 Constructor & Destructor Documentation

6.476.3.1 P0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P0< RTYPE, NA, T >::P0 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             bool lower_tail = true,
             bool log_ = false )  [inline]
```

Definition at line 128 of file dpq.h.

6.476.4 Member Function Documentation

6.476.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P0< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 132 of file dpq.h.

References Rcpp::stats::P0< RTYPE, NA, T >::log, Rcpp::stats::P0< RTYPE, NA, T >::lower, Rcpp::stats::P0< RTYPE, NA, T >::ptr, and Rcpp::stats::P0< RTYPE, NA, T >::vec.

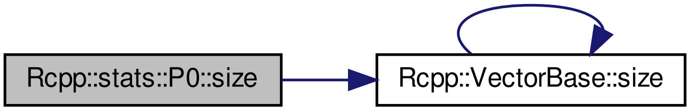
6.476.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P0< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 136 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P0< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.476.5 Member Data Documentation

6.476.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P0< RTYPE, NA, T >::log  [private]
```

Definition at line 141 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[ ]().

6.476.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P0< RTYPE, NA, T >::lower  [private]
```

Definition at line 141 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[ ]().

6.476.5.3 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::P0< RTYPE, NA, T >::ptr  [private]
```

Definition at line 139 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[ ]().

6.476.5.4 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P0< RTYPE, NA, T >::vec  [private]
```

Definition at line 140 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::P0< RTYPE, NA, T >::size().

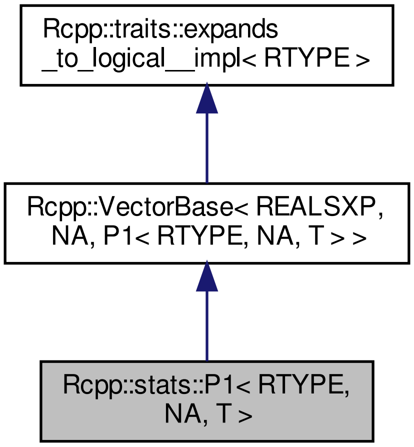
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

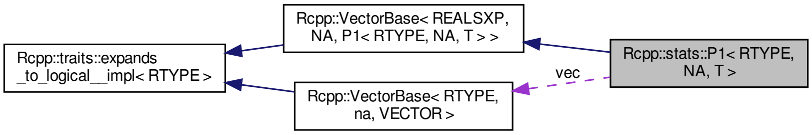
6.477 Rcpp::stats::P1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::P1< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, int, int)

Public Member Functions

- P1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- int lower
- int log

6.477.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P1< RTYPE, NA, T >

Definition at line 147 of file dpq.h.

6.477.2 Member Typedef Documentation

6.477.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::P1< RTYPE, NA, T >::FunPtr) (double, double, int, int)
```

Definition at line 150 of file dpq.h.

6.477.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P1< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 149 of file dpq.h.

6.477.3 Constructor & Destructor Documentation

6.477.3.1 P1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P1< RTYPE, NA, T >::P1 (
            FunPtr ptr_,
            const VEC_TYPE & vec_,
            double p0_,
            bool lower_tail = true,
            bool log_ = false ) [inline]
```

Definition at line 152 of file dpq.h.

6.477.4 Member Function Documentation

6.477.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P1< RTYPE, NA, T >::operator[] (
            int i ) const [inline]
```

Definition at line 156 of file dpq.h.

References Rcpp::stats::P1< RTYPE, NA, T >::log, Rcpp::stats::P1< RTYPE, NA, T >::lower, Rcpp::stats::P1< RTYPE, NA, T >::p0, Rcpp::stats::P1< RTYPE, NA, T >::ptr, and Rcpp::stats::P1< RTYPE, NA, T >::vec.

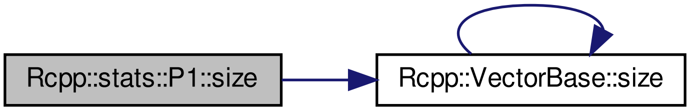
6.477.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P1< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 160 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P1< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.477.5 Member Data Documentation

6.477.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P1< RTYPE, NA, T >::log  [private]
```

Definition at line 166 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[ ]().

6.477.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P1< RTYPE, NA, T >::lower  [private]
```

Definition at line 166 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[ ]().

6.477.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P1< RTYPE, NA, T >::p0  [private]
```

Definition at line 165 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[ ]().

6.477.5.4 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::P1< RTYPE, NA, T >::ptr  [private]
```

Definition at line 163 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[ ]().

6.477.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P1< RTYPE, NA, T >::vec  [private]
```

Definition at line 164 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::P1< RTYPE, NA, T >::size().

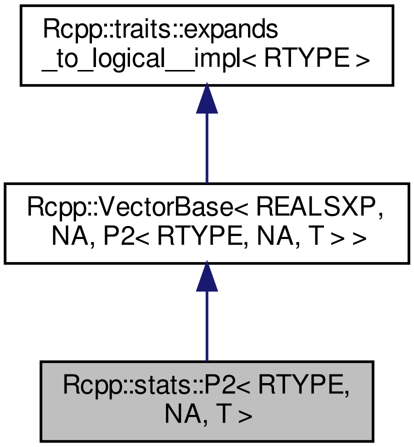
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

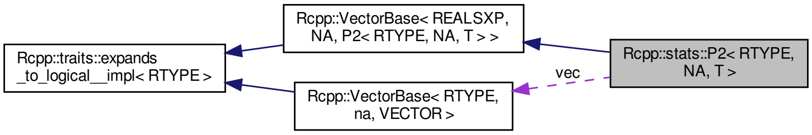
6.478 Rcpp::stats::P2< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P2< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > >
Rcpp::stats::P2< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::P2< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::P2< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, double, int, int)

Public Member Functions

- P2 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, double p1_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- double p1
- int lower
- int log

6.478.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P2< RTYPE, NA, T >

Definition at line 172 of file dpq.h.

6.478.2 Member Typedef Documentation

6.478.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(∗ Rcpp::stats::P2< RTYPE, NA, T >::FunPtr) (double, double, double, int, int)
```

Definition at line 175 of file dpq.h.

6.478.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P2< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 174 of file dpq.h.

6.478.3 Constructor & Destructor Documentation

6.478.3.1 P2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P2< RTYPE, NA, T >::P2 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             double p0_,
             double p1_,
             bool lower_tail = true,
             bool log_ = false )  [inline]
```

Definition at line 177 of file dpq.h.

6.478.4 Member Function Documentation

6.478.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P2< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 181 of file dpq.h.

References Rcpp::stats::P2< RTYPE, NA, T >::log, Rcpp::stats::P2< RTYPE, NA, T >::lower, Rcpp::stats::P2< RTYPE, NA, T >::p0, Rcpp::stats::P2< RTYPE, NA, T >::p1, Rcpp::stats::P2< RTYPE, NA, T >::ptr, and Rcpp::stats::P2< RTYPE, NA, T >::vec.

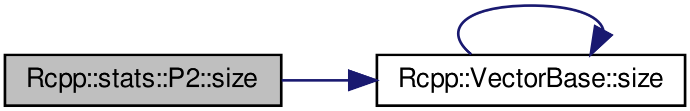
6.478.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P2< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 185 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P2< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.478.5 Member Data Documentation

6.478.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P2< RTYPE, NA, T >::log  [private]
```

Definition at line 191 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

6.478.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P2< RTYPE, NA, T >::lower  [private]
```

Definition at line 191 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

6.478.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P2< RTYPE, NA, T >::p0  [private]
```

Definition at line 190 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

6.478.5.4 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P2< RTYPE, NA, T >::p1  [private]
```

Definition at line 190 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

6.478.5.5 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::P2< RTYPE, NA, T >::ptr  [private]
```

Definition at line 188 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

6.478.5.6 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P2< RTYPE, NA, T >::vec  [private]
```

Definition at line 189 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::P2< RTYPE, NA, T >::size().

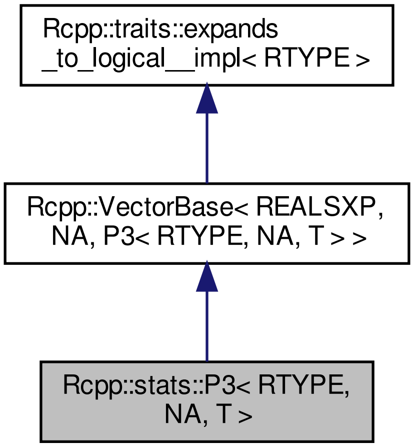
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

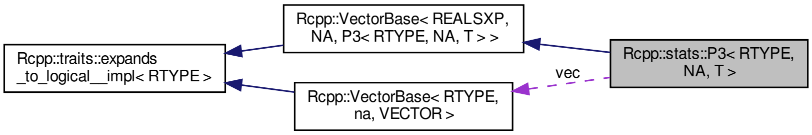
6.479 Rcpp::stats::P3< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P3< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P3< RTYPE, NA, T > >
Rcpp::stats::P3< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::P3< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, P3< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::P3< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, double, double, int, int)

Public Member Functions

- P3 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, double p1_, double p2_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- double p1
- double p2
- int lower
- int log

6.479.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::P3< RTYPE, NA, T >

Definition at line 195 of file dpq.h.

6.479.2 Member Typedef Documentation

6.479.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::P3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int,
int)
```

Definition at line 198 of file dpq.h.

6.479.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P3< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 197 of file dpq.h.

6.479.3 Constructor & Destructor Documentation

6.479.3.1 P3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P3< RTYPE, NA, T >::P3 (
            FunPtr ptr_,
            const VEC_TYPE & vec_,
            double p0_,
            double p1_,
            double p2_,
            bool lower_tail = true,
            bool log_ = false )  [inline]
```

Definition at line 200 of file dpq.h.

6.479.4 Member Function Documentation

6.479.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::operator[] (
            int i ) const  [inline]
```

Definition at line 204 of file dpq.h.

References Rcpp::stats::P3< RTYPE, NA, T >::log, Rcpp::stats::P3< RTYPE, NA, T >::lower, Rcpp::stats::P3< RTYPE, NA, T >::p0, Rcpp::stats::P3< RTYPE, NA, T >::p1, Rcpp::stats::P3< RTYPE, NA, T >::p2, Rcpp::stats::P3< RTYPE, NA, T >::ptr, and Rcpp::stats::P3< RTYPE, NA, T >::vec.

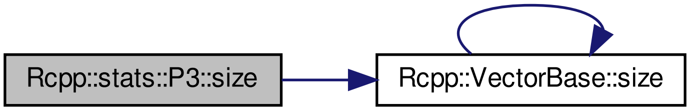
6.479.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P3< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 208 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P3< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.479.5 Member Data Documentation

6.479.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P3< RTYPE, NA, T >::log  [private]
```

Definition at line 214 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P3< RTYPE, NA, T >::lower  [private]
```

Definition at line 214 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::p0  [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.4 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::p1  [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.5 p2

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::p2  [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.6 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::P3< RTYPE, NA, T >::ptr  [private]
```

Definition at line 211 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ]().

6.479.5.7 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P3< RTYPE, NA, T >::vec  [private]
```

Definition at line 212 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::P3< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

6.480 Rcpp::attributes::Param Class Reference

Public Member Functions

- Param ()
- Param (const std::string &paramText)
- bool empty () const
- bool operator== (const Param &other) const
- bool operator!= (const Param &other) const
- const std::string & name () const
- const std::string & value () const

Private Attributes

- std::string name_
- std::string value_

6.480.1 Detailed Description

Definition at line 289 of file attributes.cpp.

6.480.2 Constructor & Destructor Documentation

6.480.2.1 Param() [1/2]

```
Rcpp::attributes::Param::Param ( )  [inline]
```

Definition at line 291 of file attributes.cpp.

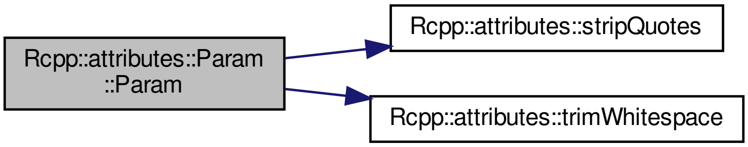
6.480.2.2 Param() [2/2]

```
Rcpp::attributes::Param::Param (
             const std::string & paramText )  [explicit]
```

Definition at line 1061 of file attributes.cpp.

References name_, Rcpp::attributes::stripQuotes(), Rcpp::attributes::trimWhitespace(), and value_.

Here is the call graph for this function:


Rcpp::attributes::Param
::Param
Rcpp::attributes::stripQuotes
Rcpp::attributes::trimWhitespace

6.480.3 Member Function Documentation

6.480.3.1 empty()

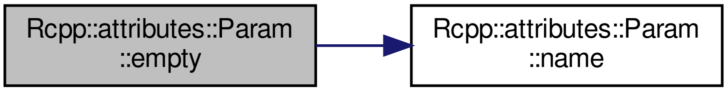
```
bool Rcpp::attributes::Param::empty ( ) const  [inline]
```

Definition at line 293 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::Attribute::hasParameter(), Rcpp::attributes::Attribute::invisible(), Rcpp::attributes↩
::operator<<(), and Rcpp::attributes::Attribute::rng().

Here is the call graph for this function:



6.480.3.2 name()

```
const std::string& Rcpp::attributes::Param::name ( ) const  [inline]
```

Definition at line 305 of file attributes.cpp.

References name_.

Referenced by empty(), and Rcpp::attributes::operator<<().

6.480.3.3 operator"!=()

```
bool Rcpp::attributes::Param::operator!= (
             const Param & other ) const  [inline]
```

Definition at line 300 of file attributes.cpp.

6.480.3.4 operator==()

```
bool Rcpp::attributes::Param::operator== (
            const Param & other ) const  [inline]
```

Definition at line 295 of file attributes.cpp.

References name_, and value_.

6.480.3.5 value()

```
const std::string& Rcpp::attributes::Param::value ( ) const  [inline]
```

Definition at line 306 of file attributes.cpp.

References value_.

Referenced by Rcpp::attributes::Attribute::customRSignature(), Rcpp::attributes::Attribute::exportedName(), Rcpp↩ ::attributes::Attribute::invisible(), Rcpp::attributes::operator<<(), and Rcpp::attributes::Attribute::rng().

6.480.4 Member Data Documentation

6.480.4.1 name_

```
std::string Rcpp::attributes::Param::name_  [private]
```

Definition at line 309 of file attributes.cpp.

Referenced by name(), operator==(), and Param().

6.480.4.2 value_

```
std::string Rcpp::attributes::Param::value_  [private]
```

Definition at line 310 of file attributes.cpp.

Referenced by operator==(), Param(), and value().

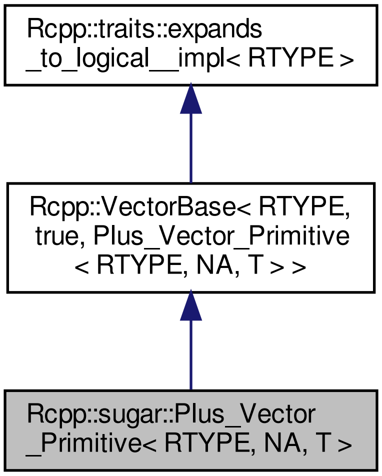
The documentation for this class was generated from the following file:

- src/attributes.cpp

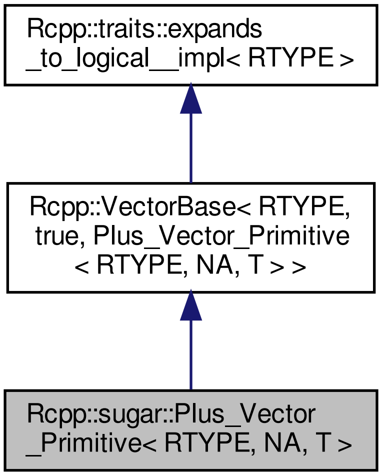
6.481 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive < RTYPE, NA, T > >
Rcpp::sugar::Plus_Vector _Primitive< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs
- bool rhs_na

6.481.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >

Definition at line 248 of file plus.h.

6.481.2 Member Typedef Documentation

6.481.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE,
NA, T >::EXT
```

Definition at line 254 of file plus.h.

6.481.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >↩
::STORAGE
```

Definition at line 252 of file plus.h.

6.481.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 251 of file plus.h.

6.481.3 Constructor & Destructor Documentation

6.481.3.1 Plus_Vector_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::Plus_Vector_Primitive (
             const VEC_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 256 of file plus.h.

6.481.4 Member Function Documentation

6.481.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 260 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.481.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 266 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.481.5 Member Data Documentation

6.481.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs  [private]
```

Definition at line 269 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::size(), Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::size(), Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::size(), and Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::size().

6.481.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs  [private]
```

Definition at line 270 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_↩ Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::operator[ ]().

6.481.5.3 rhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na  [private]
```

Definition at line 271 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Plus_Vector_↩ Primitive< RTYPE, false, T >::operator[ ]().

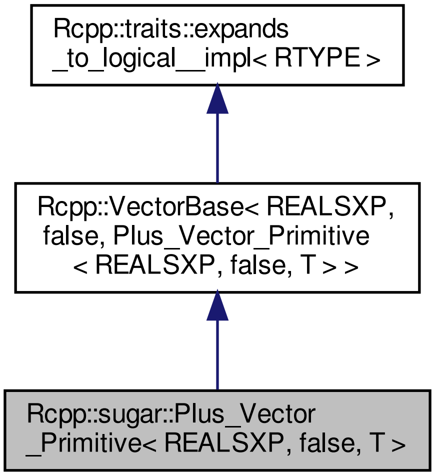
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

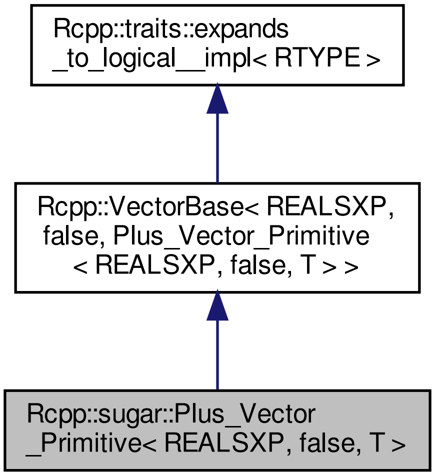
6.482 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
false, Plus_Vector_Primitive
< REALSXP, false, T > >
Rcpp::sugar::Plus_Vector
_Primitive< REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive < REALSXP, false, T > >
Rcpp::sugar::Plus_Vector _Primitive< REALSXP, false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.482.1 Detailed Description

template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >

Definition at line 323 of file plus.h.

6.482.2 Member Typedef Documentation

6.482.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor< REALSXP, false, T>::type Rcpp::sugar::Plus_Vector_Primitive<
REALSXP, false, T >::EXT
```

Definition at line 328 of file plus.h.

6.482.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 326 of file plus.h.

6.482.3 Constructor & Destructor Documentation

6.482.3.1 Plus_Vector_Primitive()

```
template<typename T >
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::Plus_Vector_Primitive (
            const VEC_TYPE & lhs_,
            double rhs_ )  [inline]
```

Definition at line 330 of file plus.h.

6.482.4 Member Function Documentation

6.482.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 333 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs.

6.482.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 337 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.482.5 Member Data Documentation

6.482.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::lhs  [private]
```

Definition at line 340 of file plus.h.

6.482.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::rhs  [private]
```

Definition at line 341 of file plus.h.

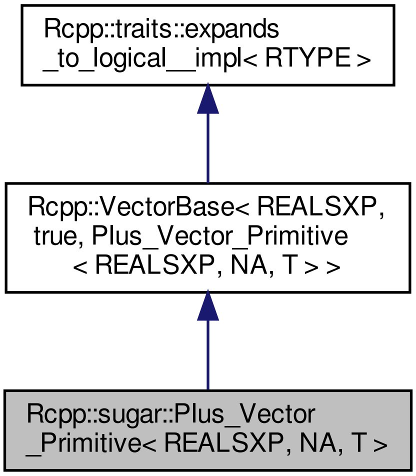
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

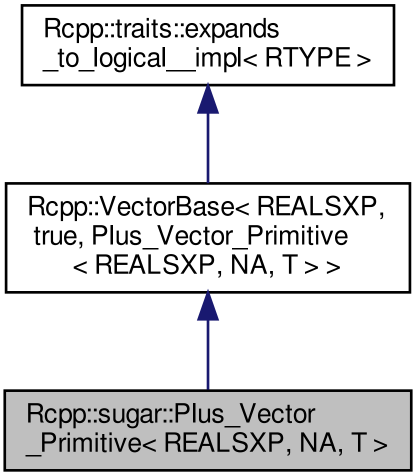
6.483 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Plus_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Plus_Vector
_Primitive< REALSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive < REALSXP, NA, T > >
Rcpp::sugar::Plus_Vector _Primitive< REALSXP, NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.483.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >

Definition at line 276 of file plus.h.

6.483.2 Member Typedef Documentation

6.483.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor< REALSXP, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive< REALSXP,
NA, T >::EXT
```

Definition at line 280 of file plus.h.

6.483.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 279 of file plus.h.

6.483.3 Constructor & Destructor Documentation

6.483.3.1 Plus_Vector_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::Plus_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 282 of file plus.h.

6.483.4 Member Function Documentation

6.483.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 286 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs.

6.483.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 290 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.483.5 Member Data Documentation

6.483.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::lhs  [private]
```

Definition at line 293 of file plus.h.

6.483.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::rhs  [private]
```

Definition at line 294 of file plus.h.

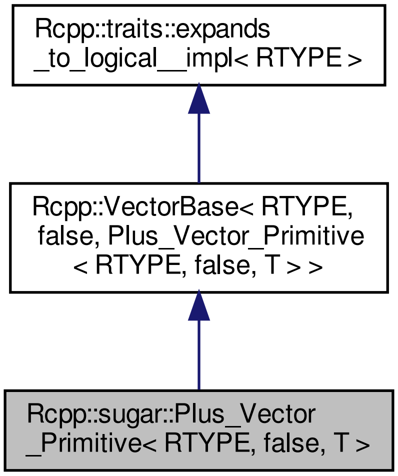
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

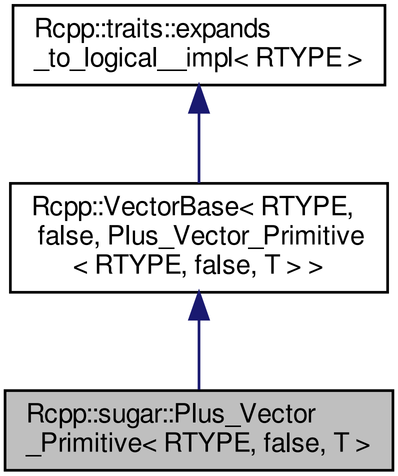
6.484 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs
- bool rhs_na

6.484.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >

Definition at line 300 of file plus.h.

6.484.2 Member Typedef Documentation

6.484.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE,
false, T >::EXT
```

Definition at line 305 of file plus.h.

6.484.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 303 of file plus.h.

6.484.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >↩
::VEC_TYPE
```

Definition at line 302 of file plus.h.

6.484.3 Constructor & Destructor Documentation

6.484.3.1 Plus_Vector_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::Plus_Vector_Primitive (
            const VEC_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 307 of file plus.h.

6.484.4 Member Function Documentation

6.484.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 310 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.484.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 314 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs.

6.484.5 Member Data Documentation

6.484.5.1 lhs

```
template<int RTYPE, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::lhs  [private]
```

Definition at line 317 of file plus.h.

6.484.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::rhs  [private]
```

Definition at line 318 of file plus.h.

6.484.5.3 rhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::rhs_na  [private]
```

Definition at line 319 of file plus.h.

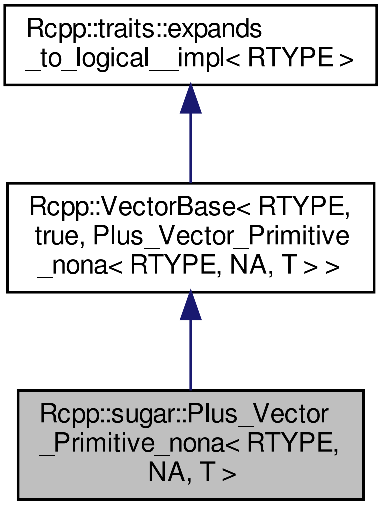
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

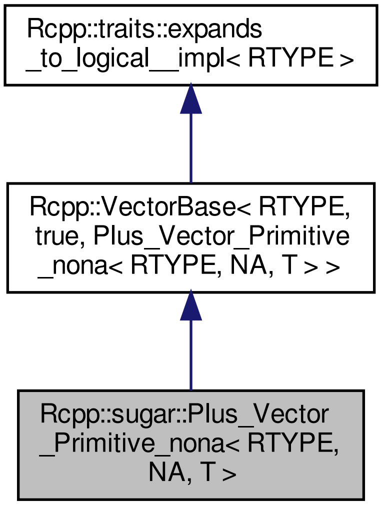
6.485 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive_nona< RTYPE, NA, T > >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive_nona (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs

6.485.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >

Definition at line 351 of file plus.h.

6.485.2 Member Typedef Documentation

6.485.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
RTYPE, NA, T >::EXT
```

Definition at line 355 of file plus.h.

6.485.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T
>::STORAGE
```

Definition at line 354 of file plus.h.

6.485.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >↩
::VEC_TYPE
```

Definition at line 353 of file plus.h.

6.485.3 Constructor & Destructor Documentation

6.485.3.1 Plus_Vector_Primitive_nona()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::Plus_Vector_Primitive_nona (
                const VEC_TYPE & lhs_,
                STORAGE rhs_ )  [inline]
```

Definition at line 357 of file plus.h.

6.485.4 Member Function Documentation

6.485.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 361 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs.

6.485.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 366 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.485.5 Member Data Documentation

6.485.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs  [private]
```

Definition at line 369 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::size(), Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::size(), Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::size(), and Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::size().

6.485.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs  [private]
```

Definition at line 370 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector↩_Primitive_nona< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >↩::operator[ ](), and Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::operator[ ]().

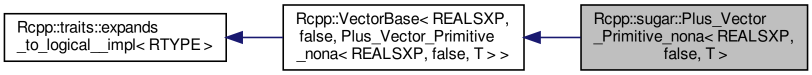
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

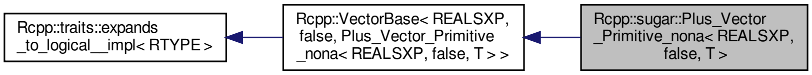
6.486 Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive _nona< REALSXP, false, T > >
Rcpp::sugar::Plus_Vector _Primitive_nona< REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive _nona< REALSXP, false, T > >
Rcpp::sugar::Plus_Vector _Primitive_nona< REALSXP, false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive_nona (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.486.1 Detailed Description

template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >

Definition at line 422 of file plus.h.

6.486.2 Member Typedef Documentation

6.486.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor< REALSXP, false, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
REALSXP, false, T >::EXT
```

Definition at line 426 of file plus.h.

6.486.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 425 of file plus.h.

6.486.3 Constructor & Destructor Documentation

6.486.3.1 Plus_Vector_Primitive_nona()

```
template<typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::Plus_Vector_Primitive_nona (
            const VEC_TYPE & lhs_,
            double rhs_ )  [inline]
```

Definition at line 428 of file plus.h.

6.486.4 Member Function Documentation

6.486.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 431 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs.

6.486.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 435 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.486.5 Member Data Documentation

6.486.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::lhs  [private]
```

Definition at line 438 of file plus.h.

6.486.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::rhs  [private]
```

Definition at line 439 of file plus.h.

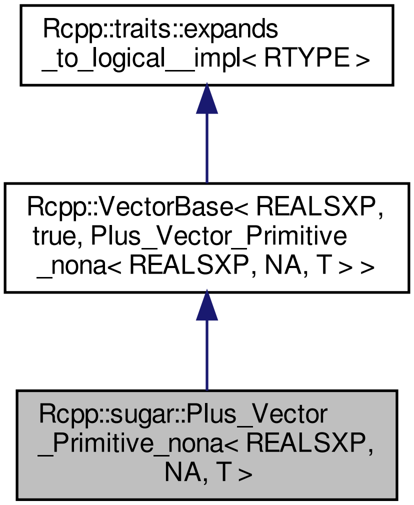
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

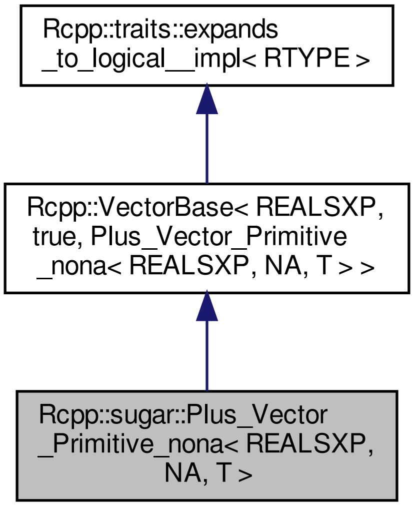
6.487 Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Plus_Vector_Primitive
_nona< REALSXP, NA, T > >
Rcpp::sugar::Plus_Vector
_Primitive_nona< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Plus_Vector_Primitive
_nona< REALSXP, NA, T > >
Rcpp::sugar::Plus_Vector
_Primitive_nona< REALSXP,
NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive_nona (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.487.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >

Definition at line 374 of file plus.h.

6.487.2 Member Typedef Documentation

6.487.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
REALSXP, NA, T >::EXT
```

Definition at line 378 of file plus.h.

6.487.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T
>::VEC_TYPE
```

Definition at line 377 of file plus.h.

6.487.3 Constructor & Destructor Documentation

6.487.3.1 Plus_Vector_Primitive_nona()

```
template<bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::Plus_Vector_Primitive_nona (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 380 of file plus.h.

6.487.4 Member Function Documentation

6.487.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 384 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_↩Primitive_nona< RTYPE, NA, T >::rhs.

6.487.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 388 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.487.5 Member Data Documentation

6.487.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::lhs  [private]
```

Definition at line 391 of file plus.h.

6.487.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::rhs  [private]
```

Definition at line 392 of file plus.h.

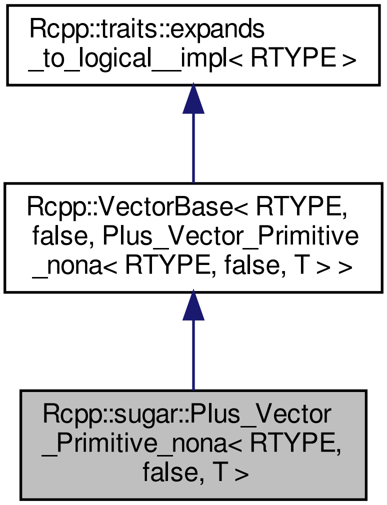
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

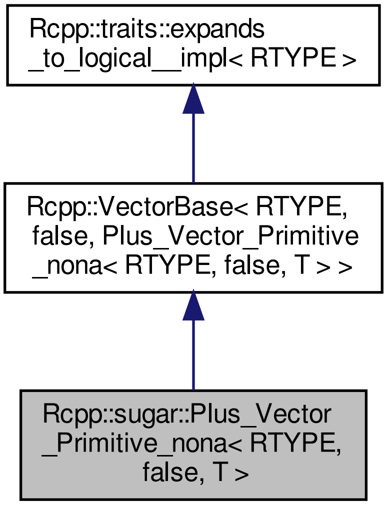
6.488 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive_nona< RTYPE, false, T > >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type EXT

Public Member Functions

- Plus_Vector_Primitive_nona (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs

6.488.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >

Definition at line 399 of file plus.h.

6.488.2 Member Typedef Documentation

6.488.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
RTYPE, false, T >::EXT
```

Definition at line 404 of file plus.h.

6.488.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false,
T >::STORAGE
```

Definition at line 402 of file plus.h.

6.488.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 401 of file plus.h.

6.488.3 Constructor & Destructor Documentation

6.488.3.1 Plus_Vector_Primitive_nona()

```
template<int RTYPE, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::Plus_Vector_Primitive_nona (
             const VEC_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 406 of file plus.h.

6.488.4 Member Function Documentation

6.488.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 409 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Plus_Vector_↩ Primitive_nona< RTYPE, NA, T >::rhs.

6.488.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 413 of file plus.h.

References Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.488.5 Member Data Documentation

6.488.5.1 lhs

```
template<int RTYPE, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::lhs  [private]
```

Definition at line 416 of file plus.h.

6.488.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::rhs  [private]
```

Definition at line 417 of file plus.h.

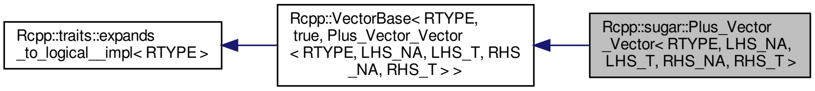
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

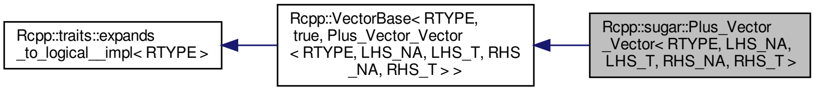
6.489 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, LHS_NA, LHS_T, RHS _NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, LHS_NA, LHS_T, RHS _NA, RHS_T > >
Rcpp::sugar::Plus_Vector _Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.489.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file plus.h.

6.489.2 Member Typedef Documentation

6.489.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file plus.h.

6.489.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file plus.h.

6.489.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file plus.h.

6.489.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file plus.h.

6.489.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 31 of file plus.h.

6.489.3 Constructor & Destructor Documentation

6.489.3.1 Plus_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 38 of file plus.h.

6.489.4 Member Function Documentation

6.489.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 41 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar↩ ::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.489.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.489.5 Member Data Documentation

6.489.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 51 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size().

6.489.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 52 of file plus.h.

Referenced by Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), and Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ]().

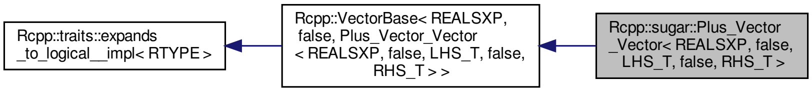
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

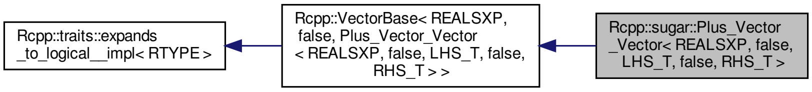
6.490 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
false, Plus_Vector_Vector
< REALSXP, false, LHS_T, false,
RHS_T > >
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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
false, Plus_Vector_Vector
< REALSXP, false, LHS_T, false,
RHS_T > >
Rcpp::sugar::Plus_Vector
_Vector< REALSXP, false,
LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.490.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Definition at line 218 of file plus.h.

6.490.2 Member Typedef Documentation

6.490.2.1 LHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 224 of file plus.h.

6.490.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 221 of file plus.h.

6.490.2.3 RHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 225 of file plus.h.

6.490.2.4 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 222 of file plus.h.

6.490.3 Constructor & Destructor Documentation

6.490.3.1 Plus_Vector_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Plus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 227 of file plus.h.

6.490.4 Member Function Documentation

6.490.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 230 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.490.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 234 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.490.5 Member Data Documentation

6.490.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 237 of file plus.h.

6.490.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 238 of file plus.h.

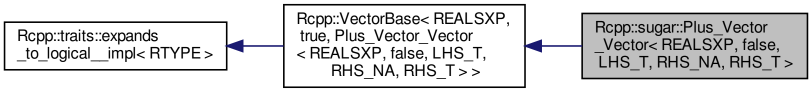
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

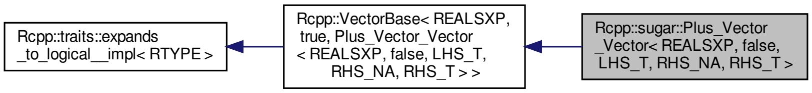
6.491 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector < REALSXP, false, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector < REALSXP, false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Plus_Vector _Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.491.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Definition at line 113 of file plus.h.

6.491.2 Member Typedef Documentation

6.491.2.1 LHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 119 of file plus.h.

6.491.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 116 of file plus.h.

6.491.2.3 RHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 120 of file plus.h.

6.491.2.4 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 117 of file plus.h.

6.491.3 Constructor & Destructor Documentation

6.491.3.1 Plus_Vector_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 122 of file plus.h.

6.491.4 Member Function Documentation

6.491.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 125 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar↩
::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.491.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 129 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.491.5 Member Data Documentation

6.491.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 132 of file plus.h.

6.491.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 133 of file plus.h.

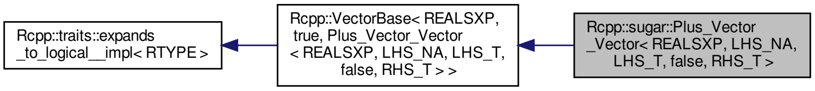
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

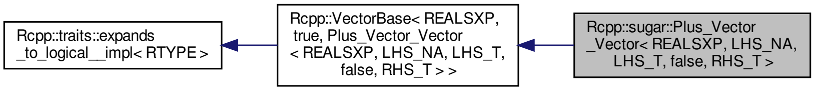
6.492 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.492.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Definition at line 166 of file plus.h.

6.492.2 Member Typedef Documentation

6.492.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 172 of file plus.h.

6.492.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 169 of file plus.h.

6.492.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 173 of file plus.h.

6.492.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 170 of file plus.h.

6.492.3 Constructor & Destructor Documentation

6.492.3.1 Plus_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Plus_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 175 of file plus.h.

6.492.4 Member Function Documentation

6.492.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 178 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.492.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 182 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.492.5 Member Data Documentation

6.492.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 185 of file plus.h.

6.492.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 186 of file plus.h.

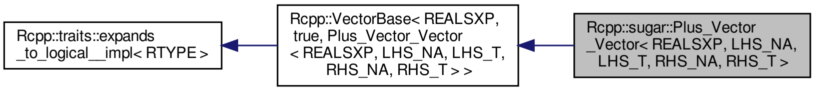
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

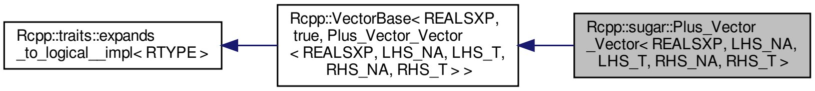
6.493 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Plus_Vector _Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.493.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 59 of file plus.h.

6.493.2 Member Typedef Documentation

6.493.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 65 of file plus.h.

6.493.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 62 of file plus.h.

6.493.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 66 of file plus.h.

6.493.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 63 of file plus.h.

6.493.3 Constructor & Destructor Documentation

6.493.3.1 Plus_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 68 of file plus.h.

6.493.4 Member Function Documentation

6.493.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 71 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.493.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 75 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.493.5 Member Data Documentation

6.493.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 78 of file plus.h.

6.493.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 79 of file plus.h.

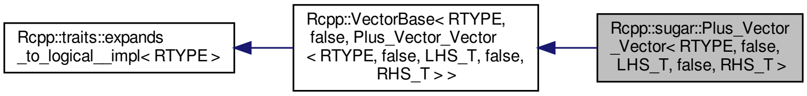
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

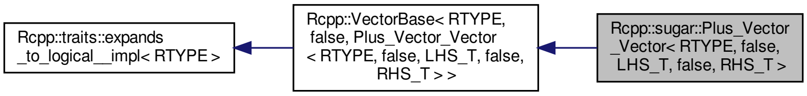
6.494 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:



Collaboration diagram for Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:



Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.494.1 Detailed Description

template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Definition at line 194 of file plus.h.

6.494.2 Member Typedef Documentation

6.494.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 200 of file plus.h.

6.494.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::LHS_TYPE
```

Definition at line 197 of file plus.h.

6.494.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 201 of file plus.h.

6.494.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::RHS_TYPE
```

Definition at line 198 of file plus.h.

6.494.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 196 of file plus.h.

6.494.3 Constructor & Destructor Documentation

6.494.3.1 Plus_Vector_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Plus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 203 of file plus.h.

6.494.4 Member Function Documentation

6.494.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 206 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar↩::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.494.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 210 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.494.5 Member Data Documentation

6.494.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 213 of file plus.h.

6.494.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 214 of file plus.h.

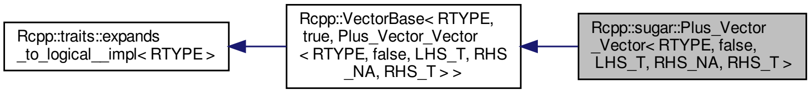
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

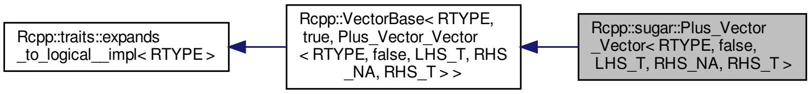
6.495 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, false, LHS_T, RHS _NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, false, LHS_T, RHS _NA, RHS_T > >
Rcpp::sugar::Plus_Vector _Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.495.1 Detailed Description

template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Definition at line 87 of file plus.h.

6.495.2 Member Typedef Documentation

6.495.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 93 of file plus.h.

6.495.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 90 of file plus.h.

6.495.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 94 of file plus.h.

6.495.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS←↩
_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 91 of file plus.h.

6.495.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 89 of file plus.h.

6.495.3 Constructor & Destructor Documentation

6.495.3.1 Plus_Vector_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 96 of file plus.h.

6.495.4 Member Function Documentation

6.495.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 99 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar←↩
::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.495.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 105 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.495.5 Member Data Documentation

6.495.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 108 of file plus.h.

6.495.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 109 of file plus.h.

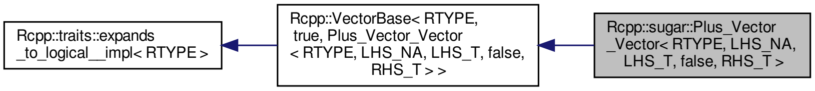
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

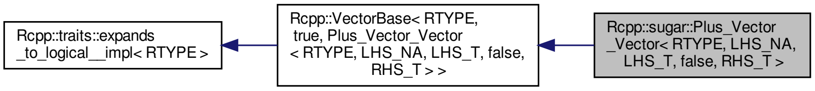
6.496 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Plus_Vector _Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Plus_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.496.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Definition at line 140 of file plus.h.

6.496.2 Member Typedef Documentation

6.496.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 146 of file plus.h.

6.496.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 143 of file plus.h.

6.496.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 147 of file plus.h.

6.496.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS↩
_T, false, RHS_T >::RHS_TYPE
```

Definition at line 144 of file plus.h.

6.496.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 142 of file plus.h.

6.496.3 Constructor & Destructor Documentation

6.496.3.1 Plus_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Plus_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 149 of file plus.h.

6.496.4 Member Function Documentation

6.496.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 152 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar↩
::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.496.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 158 of file plus.h.

References Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.496.5 Member Data Documentation

6.496.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 161 of file plus.h.

6.496.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 162 of file plus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

6.497 Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference

6.497.1 Detailed Description

template<int RTYPE, bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA >

Definition at line 28 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.498 Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA > Struct Template Reference

```
#include <pmax.h>
```

Public Member Functions

- int operator() (int left, int right) const

6.498.1 Detailed Description

template<bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >

Definition at line 57 of file pmax.h.

6.498.2 Member Function Documentation

6.498.2.1 operator()()

```
template<bool LHS_NA, bool RHS_NA>
int Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >::operator() (
             int left,
             int right ) const  [inline]
```

Definition at line 58 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.499 Rcpp::sugar::pmax_op< REALSXP, false, false > Struct Reference

```
#include <pmax.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.499.1 Detailed Description

Definition at line 48 of file pmax.h.

6.499.2 Member Function Documentation

6.499.2.1 operator()()

```
double Rcpp::sugar::pmax_op< REALSXP, false, false >::operator() (
             double left,
             double right ) const  [inline]
```

Definition at line 49 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.500 Rcpp::sugar::pmax_op< REALSXP, false, true > Struct Reference

```
#include <pmax.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.500.1 Detailed Description

Definition at line 43 of file pmax.h.

6.500.2 Member Function Documentation

6.500.2.1 operator()()

```
double Rcpp::sugar::pmax_op< REALSXP, false, true >::operator() (
            double left,
            double right ) const  [inline]
```

Definition at line 44 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.501 Rcpp::sugar::pmax_op< REALSXP, true, false > Struct Reference

```
#include <pmax.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.501.1 Detailed Description

Definition at line 38 of file pmax.h.

6.501.2 Member Function Documentation

6.501.2.1 operator()()

```
double Rcpp::sugar::pmax_op< REALSXP, true, false >::operator() (
            double left,
            double right ) const  [inline]
```

Definition at line 39 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.502 Rcpp::sugar::pmax_op< REALSXP, true, true > Struct Reference

```
#include <pmax.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.502.1 Detailed Description

Definition at line 33 of file pmax.h.

6.502.2 Member Function Documentation

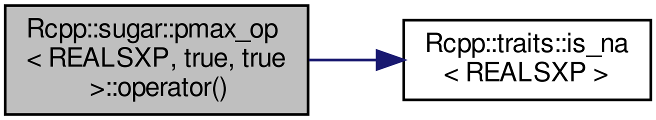
6.502.2.1 operator()()

```
double Rcpp::sugar::pmax_op< REALSXP, true, true >::operator() (
             double left,
             double right ) const  [inline]
```

Definition at line 34 of file pmax.h.

References Rcpp::traits::is_na< REALSXP >().

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.503 Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA > Class Template Reference

```
#include <pmax.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- pmax_op_Vector_Primitive (STORAGE right_)
- STORAGE operator() (STORAGE left) const

Private Attributes

- STORAGE right

6.503.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >

Definition at line 65 of file pmax.h.

6.503.2 Member Typedef Documentation

6.503.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA
>::STORAGE
```

Definition at line 67 of file pmax.h.

6.503.3 Constructor & Destructor Documentation

6.503.3.1 pmax_op_Vector_Primitive()

```
template<int RTYPE, bool NA>
Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::pmax_op_Vector_Primitive (
             STORAGE right_ )  [inline]
```

Definition at line 69 of file pmax.h.

6.503.4 Member Function Documentation

6.503.4.1 operator()()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::operator() (
             STORAGE left ) const  [inline]
```

Definition at line 71 of file pmax.h.

References Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::right.

6.503.5 Member Data Documentation

6.503.5.1 right

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::right  [private]
```

Definition at line 76 of file pmax.h.

Referenced by Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::operator()(), and Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.504 Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true > Class Reference

```
#include <pmax.h>
```

Public Member Functions

- pmax_op_Vector_Primitive (double right_)
- double operator() (double left) const

Private Attributes

- double right

6.504.1 Detailed Description

Definition at line 79 of file pmax.h.

6.504.2 Constructor & Destructor Documentation

6.504.2.1 pmax_op_Vector_Primitive()

```
Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::pmax_op_Vector_Primitive (
            double right_ )  [inline]
```

Definition at line 81 of file pmax.h.

6.504.3 Member Function Documentation

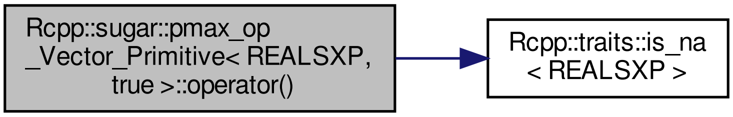
6.504.3.1 operator()()

```
double Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::operator() (
            double left ) const  [inline]
```

Definition at line 83 of file pmax.h.

References Rcpp::traits::is_na< REALSXP >(), and Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::right.

Here is the call graph for this function:



6.504.4 Member Data Documentation

6.504.4.1 right

```
double Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::right  [private]
```

Definition at line 88 of file pmax.h.

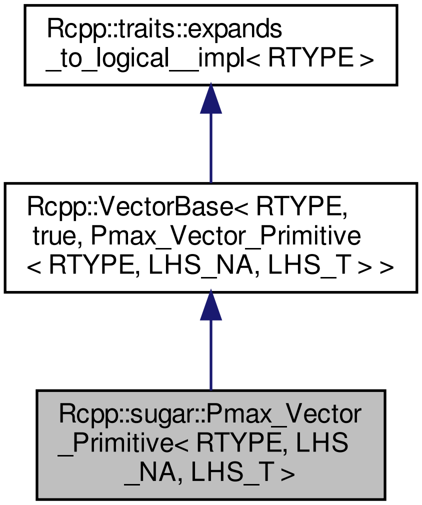
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

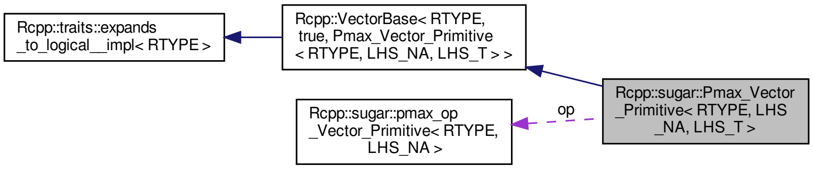
6.505 Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#include <pmax.h>
```

Inheritance diagram for Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >:



Collaboration diagram for Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >:



Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef pmax_op_Vector_Primitive< RTYPE, LHS_NA > OPERATOR

Public Member Functions

- Pmax_Vector_Primitive (const LHS_T &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_T & lhs
- OPERATOR op

6.505.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >

Definition at line 127 of file pmax.h.

6.505.2 Member Typedef Documentation

6.505.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef pmax_op_Vector_Primitive<RTYPE,LHS_NA> Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA,
LHS_T >::OPERATOR
```

Definition at line 134 of file pmax.h.

6.505.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_↩
NA, LHS_T >::STORAGE
```

Definition at line 133 of file pmax.h.

6.505.3 Constructor & Destructor Documentation

6.505.3.1 Pmax_Vector_Primitive()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::Pmax_Vector_Primitive (
            const LHS_T & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 136 of file pmax.h.

6.505.4 Member Function Documentation

6.505.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 138 of file pmax.h.

References Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs, and Rcpp::sugar::Pmax_Vector↩_Primitive< RTYPE, LHS_NA, LHS_T >::op.

6.505.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 141 of file pmax.h.

References Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs.

6.505.5 Member Data Documentation

6.505.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
const LHS_T& Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 144 of file pmax.h.

Referenced by Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[ ](), and Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size().

6.505.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
OPERATOR Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::op  [private]
```

Definition at line 145 of file pmax.h.

Referenced by Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[ ]().

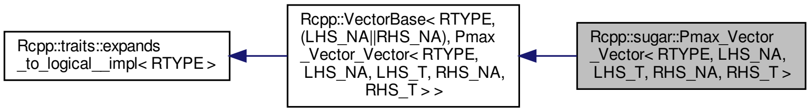
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

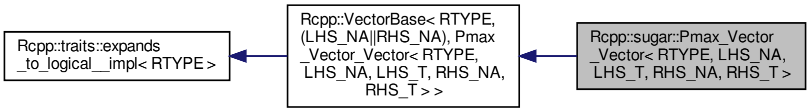
6.506 Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <pmax.h>
```

Inheritance diagram for Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), Pmax _Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), Pmax _Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Pmax_Vector _Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef pmax_op< RTYPE, LHS_NA, RHS_NA > OPERATOR

Public Member Functions

- Pmax_Vector_Vector (const LHS_T &lhs_, const RHS_T &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_T & lhs
- const RHS_T & rhs
- OPERATOR op

6.506.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 99 of file pmax.h.

6.506.2 Member Typedef Documentation

6.506.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef pmax_op<RTYPE,LHS_NA,RHS_NA> Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS←↩
_NA, RHS_T >::OPERATOR
```

Definition at line 106 of file pmax.h.

6.506.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 105 of file pmax.h.

6.506.3 Constructor & Destructor Documentation

6.506.3.1 Pmax_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Pmax_Vector_Vector (
                const LHS_T & lhs_,
                const RHS_T & rhs_ )  [inline]
```

Definition at line 108 of file pmax.h.

6.506.4 Member Function Documentation

6.506.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 110 of file pmax.h.

References Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.506.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 113 of file pmax.h.

References Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.506.5 Member Data Documentation

6.506.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_T& Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 116 of file pmax.h.

Referenced by Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[](), and Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size().

6.506.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
OPERATOR Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op  [private]
```

Definition at line 118 of file pmax.h.

Referenced by Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

6.506.5.3 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_T& Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 117 of file pmax.h.

Referenced by Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

6.507 Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference

6.507.1 Detailed Description

template<int RTYPE, bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA >

Definition at line 28 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.508 Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA > Struct Template Reference

```
#include <pmin.h>
```

Public Member Functions

- int operator() (int left, int right) const

6.508.1 Detailed Description

template<bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >

Definition at line 57 of file pmin.h.

6.508.2 Member Function Documentation

6.508.2.1 operator()()

```
template<bool LHS_NA, bool RHS_NA>
int Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >::operator() (
            int left,
            int right ) const  [inline]
```

Definition at line 58 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.509 Rcpp::sugar::pmin_op< REALSXP, false, false > Struct Reference

```
#include <pmin.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.509.1 Detailed Description

Definition at line 48 of file pmin.h.

6.509.2 Member Function Documentation

6.509.2.1 operator()()

```
double Rcpp::sugar::pmin_op< REALSXP, false, false >::operator() (
            double left,
            double right ) const  [inline]
```

Definition at line 49 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.510 Rcpp::sugar::pmin_op< REALSXP, false, true > Struct Reference

```
#include <pmin.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.510.1 Detailed Description

Definition at line 43 of file pmin.h.

6.510.2 Member Function Documentation

6.510.2.1 operator()()

```
double Rcpp::sugar::pmin_op< REALSXP, false, true >::operator() (
            double left,
            double right ) const  [inline]
```

Definition at line 44 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.511 Rcpp::sugar::pmin_op< REALSXP, true, false > Struct Reference

```
#include <pmin.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.511.1 Detailed Description

Definition at line 38 of file pmin.h.

6.511.2 Member Function Documentation

6.511.2.1 operator()()

```
double Rcpp::sugar::pmin_op< REALSXP, true, false >::operator() (
             double left,
             double right ) const  [inline]
```

Definition at line 39 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.512 Rcpp::sugar::pmin_op< REALSXP, true, true > Struct Reference

```
#include <pmin.h>
```

Public Member Functions

- double operator() (double left, double right) const

6.512.1 Detailed Description

Definition at line 33 of file pmin.h.

6.512.2 Member Function Documentation

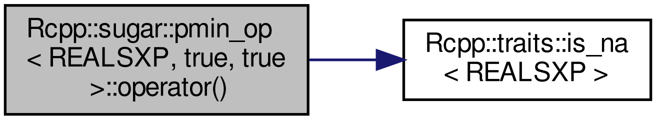
6.512.2.1 operator()()

```
double Rcpp::sugar::pmin_op< REALSXP, true, true >::operator() (
             double left,
             double right ) const  [inline]
```

Definition at line 34 of file pmin.h.

References Rcpp::traits::is_na< REALSXP >().

Here is the call graph for this function:


Rcpp::sugar::pmin_op < REALSXP, true, true >::operator()
Rcpp::traits::is_na < REALSXP >

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.513 Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA > Class Template Reference

```
#include <pmin.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- pmin_op_Vector_Primitive (STORAGE right_)
- STORAGE operator() (STORAGE left) const

Private Attributes

- STORAGE right

6.513.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >

Definition at line 65 of file pmin.h.

6.513.2 Member Typedef Documentation

6.513.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA
>::STORAGE
```

Definition at line 67 of file pmin.h.

6.513.3 Constructor & Destructor Documentation

6.513.3.1 pmin_op_Vector_Primitive()

```
template<int RTYPE, bool NA>
Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::pmin_op_Vector_Primitive (
              STORAGE right_ )  [inline]
```

Definition at line 69 of file pmin.h.

6.513.4 Member Function Documentation

6.513.4.1 operator()()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::operator() (
            STORAGE left ) const  [inline]
```

Definition at line 71 of file pmin.h.

References Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::right.

6.513.5 Member Data Documentation

6.513.5.1 right

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::right  [private]
```

Definition at line 76 of file pmin.h.

Referenced by Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::operator()(), and Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.514 Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true > Class Reference

```
#include <pmin.h>
```

Public Member Functions

- pmin_op_Vector_Primitive (double right_)
- double operator() (double left) const

Private Attributes

- double right

6.514.1 Detailed Description

Definition at line 79 of file pmin.h.

6.514.2 Constructor & Destructor Documentation

6.514.2.1 pmin_op_Vector_Primitive()

```
Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::pmin_op_Vector_Primitive (
            double right_ )  [inline]
```

Definition at line 81 of file pmin.h.

6.514.3 Member Function Documentation

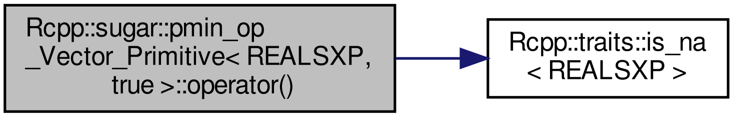
6.514.3.1 operator()()

```
double Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::operator() (
            double left ) const  [inline]
```

Definition at line 83 of file pmin.h.

References Rcpp::traits::is_na< REALSXP >(), and Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::right.

Here is the call graph for this function:



6.514.4 Member Data Documentation

6.514.4.1 right

```
double Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::right  [private]
```

Definition at line 88 of file pmin.h.

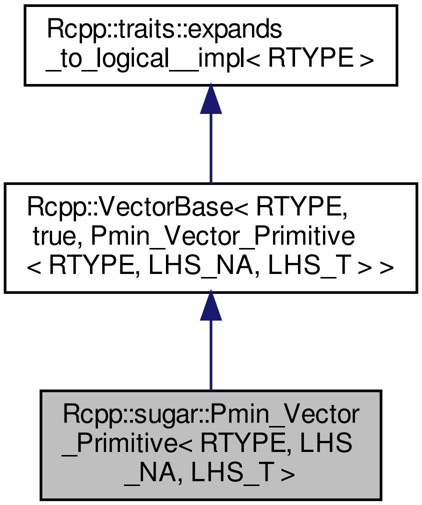
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

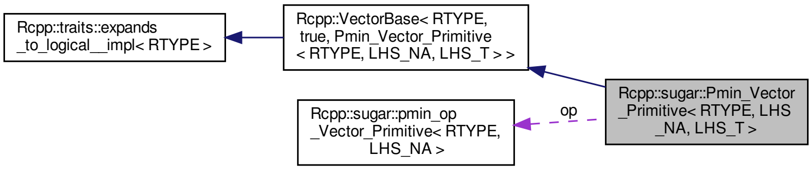
6.515 Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#include <pmin.h>
```

Inheritance diagram for Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Pmin_Vector_Primitive < RTYPE, LHS_NA, LHS_T > >
Rcpp::sugar::Pmin_Vector _Primitive< RTYPE, LHS _NA, LHS_T >

Collaboration diagram for Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Pmin_Vector_Primitive < RTYPE, LHS_NA, LHS_T > >
Rcpp::sugar::Pmin_Vector _Primitive< RTYPE, LHS _NA, LHS_T >
Rcpp::sugar::pmin_op _Vector_Primitive< RTYPE, LHS_NA >
op

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef pmin_op_Vector_Primitive< RTYPE, LHS_NA > OPERATOR

Public Member Functions

- Pmin_Vector_Primitive (const LHS_T &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_T & lhs
- OPERATOR op

6.515.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >

Definition at line 127 of file pmin.h.

6.515.2 Member Typedef Documentation

6.515.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef pmin_op_Vector_Primitive<RTYPE,LHS_NA> Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA,
LHS_T >::OPERATOR
```

Definition at line 134 of file pmin.h.

6.515.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_↩
NA, LHS_T >::STORAGE
```

Definition at line 133 of file pmin.h.

6.515.3 Constructor & Destructor Documentation

6.515.3.1 Pmin_Vector_Primitive()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::Pmin_Vector_Primitive (
             const LHS_T & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 136 of file pmin.h.

6.515.4 Member Function Documentation

6.515.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 138 of file pmin.h.

References Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs, and Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::op.

6.515.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 139 of file pmin.h.

References Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs.

6.515.5 Member Data Documentation

6.515.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
const LHS_T& Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs  [private]
```

Definition at line 142 of file pmin.h.

Referenced by Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[ ](), and Rcpp::sugar::↩ Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size().

6.515.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
OPERATOR Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::op  [private]
```

Definition at line 143 of file pmin.h.

Referenced by Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[ ]().

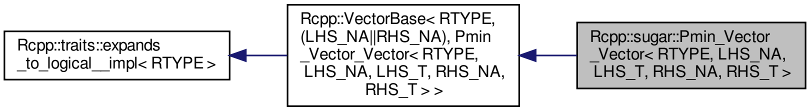
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

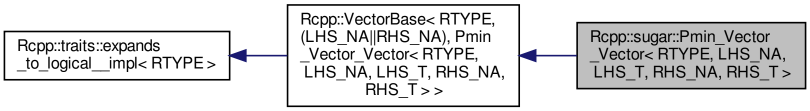
6.516 Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <pmin.h>
```

Inheritance diagram for Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), Pmin
_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, (LHS_NA||RHS_NA), Pmin
_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Pmin_Vector
_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef pmin_op< RTYPE, LHS_NA, RHS_NA > OPERATOR

Public Member Functions

- Pmin_Vector_Vector (const LHS_T &lhs_, const RHS_T &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_T & lhs
- const RHS_T & rhs
- OPERATOR op

6.516.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 99 of file pmin.h.

6.516.2 Member Typedef Documentation

6.516.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef pmin_op<RTYPE,LHS_NA,RHS_NA> Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS↩
_NA, RHS_T >::OPERATOR
```

Definition at line 106 of file pmin.h.

6.516.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 105 of file pmin.h.

6.516.3 Constructor & Destructor Documentation

6.516.3.1 Pmin_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Pmin_Vector_Vector (
             const LHS_T & lhs_,
             const RHS_T & rhs_ )  [inline]
```

Definition at line 108 of file pmin.h.

6.516.4 Member Function Documentation

6.516.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 110 of file pmin.h.

References Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, Rcpp::sugar↩ ::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op, and Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.516.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 113 of file pmin.h.

References Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.516.5 Member Data Documentation

6.516.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_T& Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 116 of file pmin.h.

Referenced by Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), and Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size().

6.516.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
OPERATOR Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op  [private]
```

Definition at line 118 of file pmin.h.

Referenced by Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

6.516.5.3 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_T& Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 117 of file pmin.h.

Referenced by Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

6.517 Rcpp::traits::pointer_wrap_tag Struct Reference

```
#include <module_wrap_traits.h>
```

6.517.1 Detailed Description

Definition at line 31 of file module_wrap_traits.h.

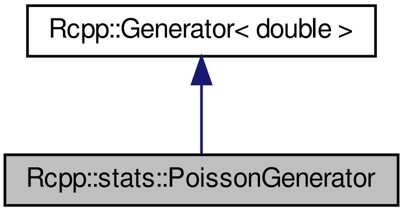
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

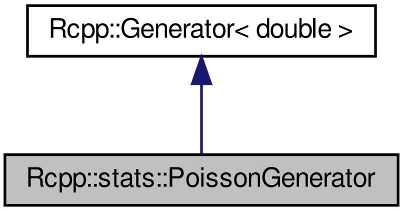
6.518 Rcpp::stats::PoissonGenerator Class Reference

```
#include <rpois.h>
```

Inheritance diagram for Rcpp::stats::PoissonGenerator:



Collaboration diagram for Rcpp::stats::PoissonGenerator:



Public Member Functions

- PoissonGenerator (double mu_)
- double operator() () const

Private Attributes

- double mu

Additional Inherited Members

6.518.1 Detailed Description

Definition at line 28 of file rpois.h.

6.518.2 Constructor & Destructor Documentation

6.518.2.1 PoissonGenerator()

```
Rcpp::stats::PoissonGenerator::PoissonGenerator (
            double mu_ )  [inline]
```

Definition at line 30 of file rpois.h.

6.518.3 Member Function Documentation

6.518.3.1 operator()()

```
double Rcpp::stats::PoissonGenerator::operator() ( ) const  [inline]
```

Definition at line 31 of file rpois.h.

References mu.

6.518.4 Member Data Documentation

6.518.4.1 mu

```
double Rcpp::stats::PoissonGenerator::mu  [private]
```

Definition at line 33 of file rpois.h.

Referenced by operator()().

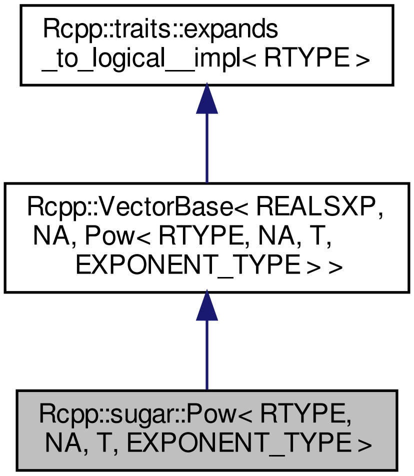
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rpois.h

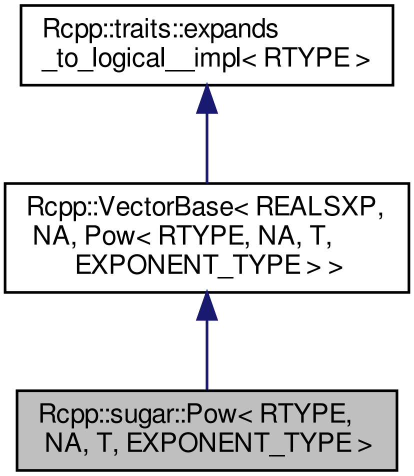
6.519 Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > Class Template Reference

```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Pow< RTYPE, NA, T, EXPONENT_TYPE > >
Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >

Collaboration diagram for Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >:



Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Pow (const T &object_, EXPONENT_TYPE exponent)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T & object
- EXPONENT_TYPE op

6.519.1 Detailed Description

template<int RTYPE, bool NA, typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >

Definition at line 29 of file pow.h.

6.519.2 Member Typedef Documentation

6.519.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE
>::STORAGE
```

Definition at line 31 of file pow.h.

6.519.3 Constructor & Destructor Documentation

6.519.3.1 Pow()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::Pow (
            const T & object_,
            EXPONENT_TYPE exponent )  [inline]
```

Definition at line 33 of file pow.h.

6.519.4 Member Function Documentation

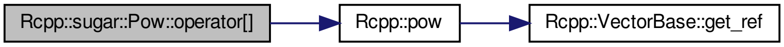
6.519.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 35 of file pow.h.

References Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op, and Rcpp::pow().

Here is the call graph for this function:



6.519.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
R_xlen_t Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::size ( ) const  [inline]
```

Definition at line 38 of file pow.h.

6.519.5 Member Data Documentation

6.519.5.1 object

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
const T& Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::object  [private]
```

Definition at line 41 of file pow.h.

6.519.5.2 op

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
EXPONENT_TYPE Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op  [private]
```

Definition at line 42 of file pow.h.

Referenced by Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::operator[ ](), Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::operator[ ](), and Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >↩ ::operator[ ]().

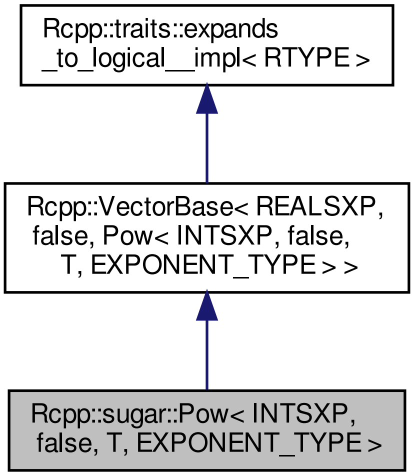
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pow.h

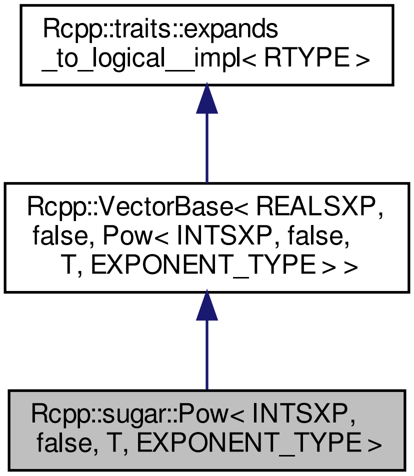
6.520 Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE > Class Template Reference

```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
false, Pow< INTSXP, false,
T, EXPONENT_TYPE > >
Rcpp::sugar::Pow< INTSXP,
false, T, EXPONENT_TYPE >

Collaboration diagram for Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
false, Pow< INTSXP, false,
T, EXPONENT_TYPE > >
Rcpp::sugar::Pow< INTSXP,
false, T, EXPONENT_TYPE >

Public Member Functions

- Pow (const T &object_, EXPONENT_TYPE exponent)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T & object
- EXPONENT_TYPE op

Additional Inherited Members

6.520.1 Detailed Description

template<typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >

Definition at line 61 of file pow.h.

6.520.2 Constructor & Destructor Documentation

6.520.2.1 Pow()

```
template<typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::Pow (
             const T & object_,
             EXPONENT_TYPE exponent ) [inline]
```

Definition at line 63 of file pow.h.

6.520.3 Member Function Documentation

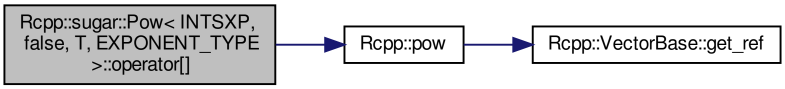
6.520.3.1 operator[]()

```
template<typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::operator[] (
             R_xlen_t i ) const [inline]
```

Definition at line 65 of file pow.h.

References Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op, and Rcpp::pow().

Here is the call graph for this function:


Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::operator[]
Rcpp::pow
Rcpp::VectorBase::get_ref

6.520.3.2 size()

```
template<typename T , typename EXPONENT_TYPE >
R_xlen_t Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::size ( ) const [inline]
```

Definition at line 68 of file pow.h.

6.520.4 Member Data Documentation

6.520.4.1 object

```
template<typename T , typename EXPONENT_TYPE >
const T& Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::object  [private]
```

Definition at line 71 of file pow.h.

6.520.4.2 op

```
template<typename T , typename EXPONENT_TYPE >
EXPONENT_TYPE Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::op  [private]
```

Definition at line 72 of file pow.h.

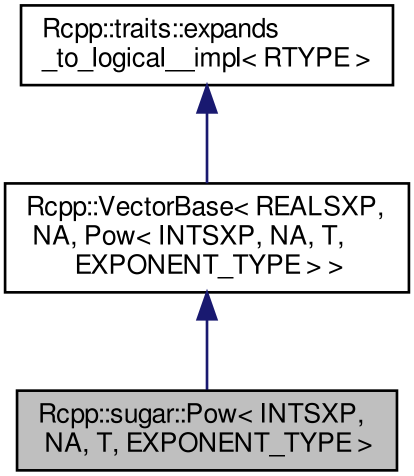
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pow.h

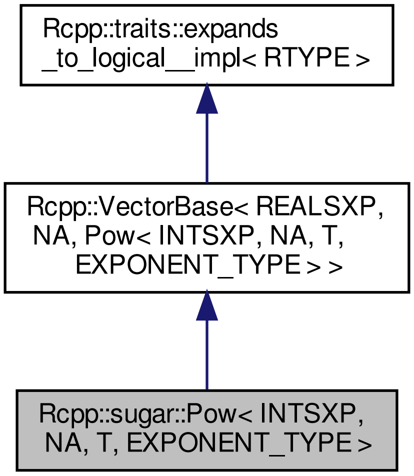
6.521 Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE > Class Template Reference

```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >:



Collaboration diagram for Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >:



Public Member Functions

- Pow (const T &object_, EXPONENT_TYPE exponent)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const T & object
- EXPONENT_TYPE op

Additional Inherited Members

6.521.1 Detailed Description

template<bool NA, typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >

Definition at line 46 of file pow.h.

6.521.2 Constructor & Destructor Documentation

6.521.2.1 Pow()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::Pow (
             const T & object_,
             EXPONENT_TYPE exponent )  [inline]
```

Definition at line 48 of file pow.h.

6.521.3 Member Function Documentation

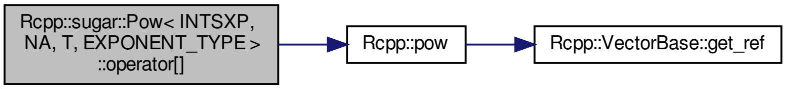
6.521.3.1 operator[]()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 50 of file pow.h.

References Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op, and Rcpp::pow().

Here is the call graph for this function:


Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE > ::operator[]
Rcpp::pow
Rcpp::VectorBase::get_ref

6.521.3.2 size()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
R_xlen_t Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::size ( ) const  [inline]
```

Definition at line 54 of file pow.h.

6.521.4 Member Data Documentation

6.521.4.1 object

```
template<bool NA, typename T , typename EXPONENT_TYPE >
const T& Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::object  [private]
```

Definition at line 57 of file pow.h.

6.521.4.2 op

```
template<bool NA, typename T , typename EXPONENT_TYPE >
EXPONENT_TYPE Rcpp::sugar::Pow< 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

6.522 Rcpp::PreserveStorage< CLASS > Class Template Reference

```
#include <PreserveStorage.h>
```

Public Member Functions

- PreserveStorage ()
- ∼PreserveStorage ()
- void set__ (SEXP x)
- SEXP get__ () const
- SEXP invalidate__ ()
- template<typename T >
  T & copy__ (const T &other)
- bool inherits (const char ∗clazz) const
- operator SEXP () const

Private Attributes

- SEXP data
- SEXP token

6.522.1 Detailed Description

template<typename CLASS>
class Rcpp::PreserveStorage< CLASS >

Definition at line 28 of file PreserveStorage.h.

6.522.2 Constructor & Destructor Documentation

6.522.2.1 PreserveStorage()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::PreserveStorage ( )  [inline]
```

Definition at line 31 of file PreserveStorage.h.

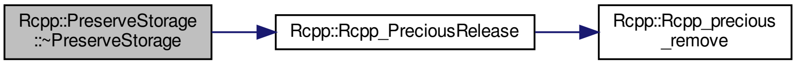
6.522.2.2 ∼PreserveStorage()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::∼PreserveStorage ( )  [inline]
```

Definition at line 33 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data, Rcpp::Rcpp_PreciousRelease(), and Rcpp::PreserveStorage< CLASS >::token.

Here is the call graph for this function:



6.522.3 Member Function Documentation

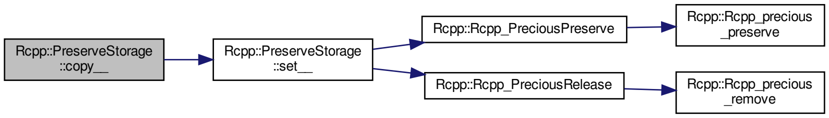
6.522.3.1 copy__()

```
template<typename CLASS >
template<typename T >
T& Rcpp::PreserveStorage< CLASS >::copy__ (
             const T & other ) [inline]
```

Definition at line 64 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::set__().

Here is the call graph for this function:


Rcpp::PreserveStorage ::copy__
Rcpp::PreserveStorage ::set__
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious _preserve
Rcpp::Rcpp_precious _remove

6.522.3.2 get__()

```
template<typename CLASS >
SEXP Rcpp::PreserveStorage< CLASS >::get__ ( ) const [inline]
```

Definition at line 51 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl().

6.522.3.3 inherits()

```
template<typename CLASS >
bool Rcpp::PreserveStorage< CLASS >::inherits (
             const char * clazz ) const [inline]
```

Definition at line 71 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

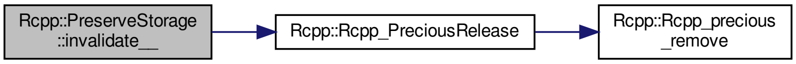
6.522.3.4 invalidate__()

```
template<typename CLASS >
SEXP Rcpp::PreserveStorage< CLASS >::invalidate__ ( )  [inline]
```

Definition at line 55 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data, Rcpp::Rcpp_PreciousRelease(), and Rcpp::PreserveStorage< CLASS >::token.

Here is the call graph for this function:



6.522.3.5 operator SEXP()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::operator SEXP ( ) const  [inline]
```

Definition at line 75 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

6.522.3.6 set__()

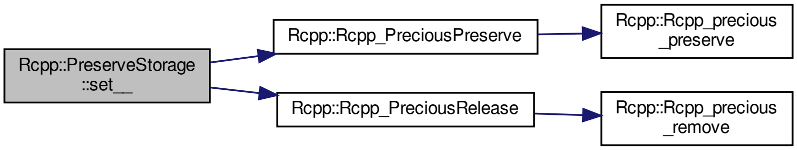
```
template<typename CLASS >
void Rcpp::PreserveStorage< CLASS >::set__ (
            SEXP x )  [inline]
```

Definition at line 39 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), and Rcpp::PreserveStorage< CLASS >::token.

Referenced by Rcpp::PreserveStorage< CLASS >::copy__().

Here is the call graph for this function:



6.522.4 Member Data Documentation

6.522.4.1 data

```
template<typename CLASS >
SEXP Rcpp::PreserveStorage< CLASS >::data  [private]
```

Definition at line 78 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::PreserveStorage< CLASS >::inherits(), Rcpp::PreserveStorage< CLASS >::invalidate__(), Rcpp::PreserveStorage< CLASS >::operator SEXP(), Rcpp::PreserveStorage< CLASS >::set__(), and Rcpp::PreserveStorage< CLASS >::~PreserveStorage().

6.522.4.2 token

```
template<typename CLASS >
SEXP Rcpp::PreserveStorage< CLASS >::token  [private]
```

Definition at line 79 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< CLASS >::invalidate__(), Rcpp::PreserveStorage< CLASS >::set__(), and Rcpp::PreserveStorage< CLASS >::~PreserveStorage().

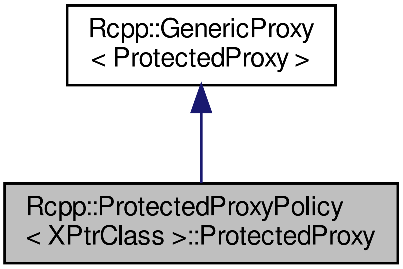
The documentation for this class was generated from the following file:

- inst/include/Rcpp/storage/PreserveStorage.h

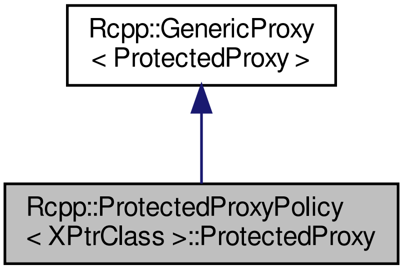
6.523 Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy Class Reference

```
#include <ProtectedProxy.h>
```

Inheritance diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy:



Collaboration diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy:



Public Member Functions

- ProtectedProxy (XPtrClass &xp_)
- template<typename U >
  ProtectedProxy & operator= (const U &u)
- template<typename U >
  operator U () const
- operator SEXP () const

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- XPtrClass & xp

6.523.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy

Definition at line 29 of file ProtectedProxy.h.

6.523.2 Constructor & Destructor Documentation

6.523.2.1 ProtectedProxy()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::ProtectedProxy (
             XPtrClass & xp_ )  [inline]
```

Definition at line 31 of file ProtectedProxy.h.

6.523.3 Member Function Documentation

6.523.3.1 get()

```
template<class XPtrClass >
SEXP Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get ( ) const  [inline], [private]
```

Definition at line 51 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator SEXP(), and Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator U().

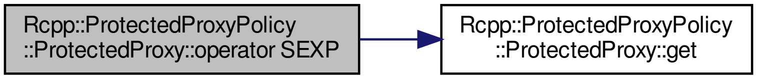
6.523.3.2 operator SEXP()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator SEXP ( ) const  [inline]
```

Definition at line 44 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get().

Here is the call graph for this function:


Rcpp::ProtectedProxyPolicy
::ProtectedProxy::operator SEXP
Rcpp::ProtectedProxyPolicy
::ProtectedProxy::get

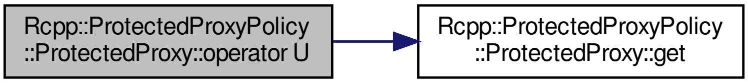
6.523.3.3 operator U()

```
template<class XPtrClass >
template<typename U >
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator U ( ) const  [inline]
```

Definition at line 40 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get().

Here is the call graph for this function:


Rcpp::ProtectedProxyPolicy
::ProtectedProxy::operator U
Rcpp::ProtectedProxyPolicy
::ProtectedProxy::get

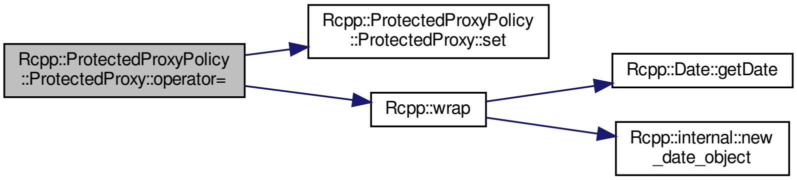
6.523.3.4 operator=()

```
template<class XPtrClass >
template<typename U >
ProtectedProxy& Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator= (
             const U & u )  [inline]
```

Definition at line 34 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set(), and Rcpp::wrap().

Here is the call graph for this function:



6.523.3.5 set()

```
template<class XPtrClass >
void Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set (
             SEXP x )  [inline], [private]
```

Definition at line 55 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator=().

6.523.4 Member Data Documentation

6.523.4.1 xp

```
template<class XPtrClass >
XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp  [private]
```

Definition at line 49 of file ProtectedProxy.h.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get(), and Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

6.524 Rcpp::ProtectedProxyPolicy< XPtrClass > Class Template Reference

```
#include <ProtectedProxy.h>
```

Classes

- class const_ProtectedProxy
- class ProtectedProxy

Public Member Functions

- ProtectedProxy prot ()
- const_ProtectedProxy prot () const

6.524.1 Detailed Description

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >

Definition at line 26 of file ProtectedProxy.h.

6.524.2 Member Function Documentation

6.524.2.1 prot() [1/2]

```
template<class XPtrClass >
ProtectedProxy Rcpp::ProtectedProxyPolicy< XPtrClass >::prot ( )  [inline]
```

Definition at line 83 of file ProtectedProxy.h.

6.524.2.2 prot() [2/2]

```
template<class XPtrClass >
const_ProtectedProxy Rcpp::ProtectedProxyPolicy< XPtrClass >::prot ( ) const  [inline]
```

Definition at line 87 of file ProtectedProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

6.525 Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_const_proxy< RTYPE, StoragePolicy >::type > type

6.525.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >

Definition at line 267 of file proxy.h.

6.525.2 Member Typedef Documentation

6.525.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_const_proxy<RTYPE, StoragePolicy>↩
::type > Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 268 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.526 Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_proxy< RTYPE, StoragePolicy >::type > type

6.526.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy >

Definition at line 259 of file proxy.h.

6.526.2 Member Typedef Documentation

6.526.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_proxy<RTYPE, StoragePolicy>::type >
Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 260 of file proxy.h.

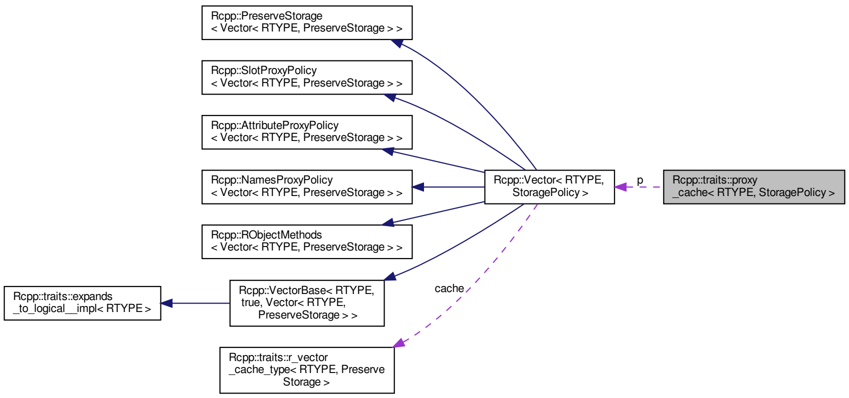
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.527 Rcpp::traits::proxy_cache< RTYPE, StoragePolicy > Class Template Reference

```
#include <traits.h>
```

Collaboration diagram for Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >
Rcpp::Vector< RTYPE, StoragePolicy >
p
cache
Rcpp::traits::proxy _cache< RTYPE, StoragePolicy >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef r_vector_iterator< RTYPE, StoragePolicy >::type iterator
- typedef r_vector_const_iterator< RTYPE, StoragePolicy >::type const_iterator
- typedef r_vector_proxy< RTYPE, StoragePolicy >::type proxy
- typedef r_vector_const_proxy< RTYPE, StoragePolicy >::type const_proxy

Public Member Functions

- proxy_cache ()
- ∼proxy_cache ()
- void update (const VECTOR &v)
- iterator get () const
- const_iterator get_const () const
- proxy ref ()
- proxy ref (R_xlen_t i)
- const_proxy ref () const
- const_proxy ref (R_xlen_t i) const

Private Attributes

- VECTOR ∗ p

6.527.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >

Definition at line 55 of file traits.h.

6.527.2 Member Typedef Documentation

6.527.2.1 const_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_const_iterator<RTYPE, StoragePolicy>::type Rcpp::traits::proxy_cache< RTYPE,
StoragePolicy >::const_iterator
```

Definition at line 59 of file traits.h.

6.527.2.2 const_proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_const_proxy<RTYPE, StoragePolicy>::type Rcpp::traits::proxy_cache< RTYPE, Storage↩
Policy >::const_proxy
```

Definition at line 61 of file traits.h.

6.527.2.3 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_iterator<RTYPE, StoragePolicy>::type Rcpp::traits::proxy_cache< RTYPE, Storage↩
Policy >::iterator
```

Definition at line 58 of file traits.h.

6.527.2.4 proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_proxy<RTYPE, StoragePolicy>::type Rcpp::traits::proxy_cache< RTYPE, Storage↩
Policy >::proxy
```

Definition at line 60 of file traits.h.

6.527.2.5 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >↩
::VECTOR
```

Definition at line 57 of file traits.h.

6.527.3 Constructor & Destructor Documentation

6.527.3.1 proxy_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::proxy_cache ( )  [inline]
```

Definition at line 63 of file traits.h.

6.527.3.2 ∼proxy_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::∼proxy_cache ( )  [inline]
```

Definition at line 64 of file traits.h.

6.527.4 Member Function Documentation

6.527.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get ( ) const  [inline]
```

Definition at line 68 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.2 get_const()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get_const ( ) const  [inline]
```

Definition at line 70 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.3 ref() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref ( )  [inline]
```

Definition at line 72 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.4 ref() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref ( ) const  [inline]
```

Definition at line 75 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.5 ref() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref (
            R_xlen_t i ) [inline]
```

Definition at line 73 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.6 ref() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref (
            R_xlen_t i ) const [inline]
```

Definition at line 76 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.4.7 update()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::update (
            const VECTOR & v ) [inline]
```

Definition at line 65 of file traits.h.

References Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p.

6.527.5 Member Data Documentation

6.527.5.1 p

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
VECTOR* Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p [private]
```

Definition at line 79 of file traits.h.

Referenced by Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get(), Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get_const(), Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref(), and Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::update().

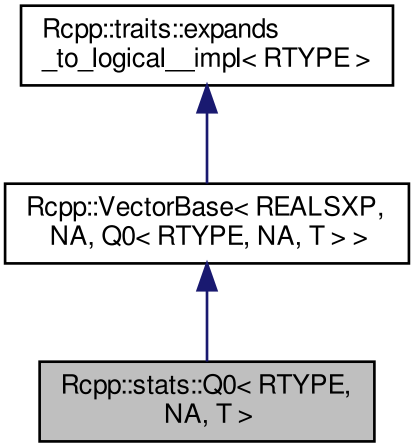
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/traits.h

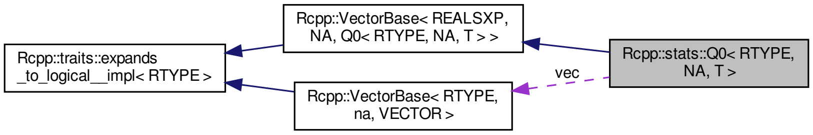
6.528 Rcpp::stats::Q0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q0< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > >
Rcpp::stats::Q0< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::Q0< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::Q0< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, int, int)

Public Member Functions

- Q0 (FunPtr ptr_, const VEC_TYPE &vec_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- int lower
- int log

6.528.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q0< RTYPE, NA, T >

Definition at line 221 of file dpq.h.

6.528.2 Member Typedef Documentation

6.528.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::Q0< RTYPE, NA, T >::FunPtr) (double, int, int)
```

Definition at line 224 of file dpq.h.

6.528.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q0< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 223 of file dpq.h.

6.528.3 Constructor & Destructor Documentation

6.528.3.1 Q0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q0< RTYPE, NA, T >::Q0 (
            FunPtr ptr_,
            const VEC_TYPE & vec_,
            bool lower_tail = true,
            bool log_ = false )  [inline]
```

Definition at line 226 of file dpq.h.

6.528.4 Member Function Documentation

6.528.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q0< RTYPE, NA, T >::operator[] (
            int i ) const  [inline]
```

Definition at line 230 of file dpq.h.

References Rcpp::stats::Q0< RTYPE, NA, T >::log, Rcpp::stats::Q0< RTYPE, NA, T >::lower, Rcpp::stats::Q0< RTYPE, NA, T >::ptr, and Rcpp::stats::Q0< RTYPE, NA, T >::vec.

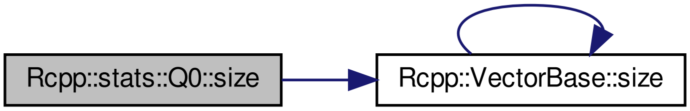
6.528.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q0< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 234 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q0< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.528.5 Member Data Documentation

6.528.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q0< RTYPE, NA, T >::log  [private]
```

Definition at line 239 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[ ]().

6.528.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q0< RTYPE, NA, T >::lower  [private]
```

Definition at line 239 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[ ]().

6.528.5.3 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::Q0< RTYPE, NA, T >::ptr  [private]
```

Definition at line 237 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[ ]().

6.528.5.4 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q0< RTYPE, NA, T >::vec  [private]
```

Definition at line 238 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::Q0< RTYPE, NA, T >::size().

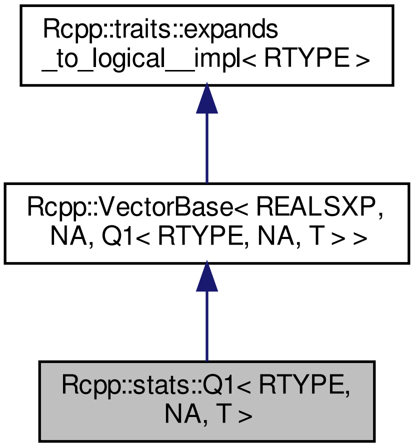
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

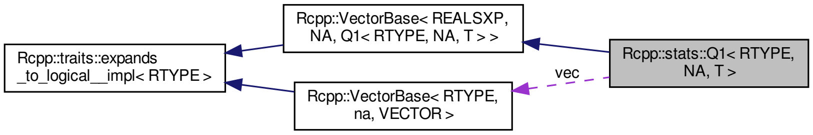
6.529 Rcpp::stats::Q1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::Q1< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, int, int)

Public Member Functions

- Q1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- int lower
- int log

6.529.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q1< RTYPE, NA, T >

Definition at line 244 of file dpq.h.

6.529.2 Member Typedef Documentation

6.529.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::Q1< RTYPE, NA, T >::FunPtr) (double, double, int, int)
```

Definition at line 247 of file dpq.h.

6.529.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q1< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 246 of file dpq.h.

6.529.3 Constructor & Destructor Documentation

6.529.3.1 Q1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q1< RTYPE, NA, T >::Q1 (
            FunPtr ptr_,
            const VEC_TYPE & vec_,
            double p0_,
            bool lower_tail = true,
            bool log_ = false )  [inline]
```

Definition at line 249 of file dpq.h.

6.529.4 Member Function Documentation

6.529.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q1< RTYPE, NA, T >::operator[] (
            int i ) const  [inline]
```

Definition at line 253 of file dpq.h.

References Rcpp::stats::Q1< RTYPE, NA, T >::log, Rcpp::stats::Q1< RTYPE, NA, T >::lower, Rcpp::stats::Q1< RTYPE, NA, T >::p0, Rcpp::stats::Q1< RTYPE, NA, T >::ptr, and Rcpp::stats::Q1< RTYPE, NA, T >::vec.

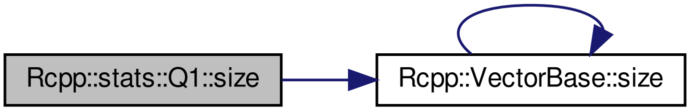
6.529.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q1< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 257 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q1< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.529.5 Member Data Documentation

6.529.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q1< RTYPE, NA, T >::log  [private]
```

Definition at line 263 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[ ]().

6.529.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q1< RTYPE, NA, T >::lower  [private]
```

Definition at line 263 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[ ]().

6.529.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q1< RTYPE, NA, T >::p0  [private]
```

Definition at line 262 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[ ]().

6.529.5.4 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::Q1< RTYPE, NA, T >::ptr  [private]
```

Definition at line 260 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[ ]().

6.529.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q1< RTYPE, NA, T >::vec  [private]
```

Definition at line 261 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::Q1< RTYPE, NA, T >::size().

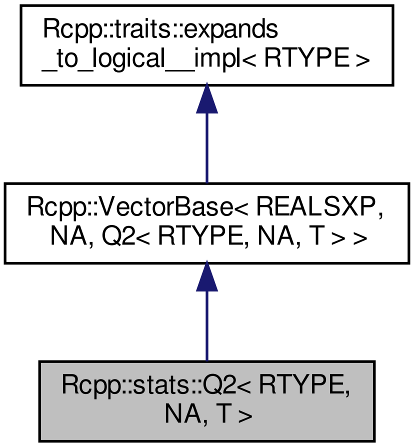
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

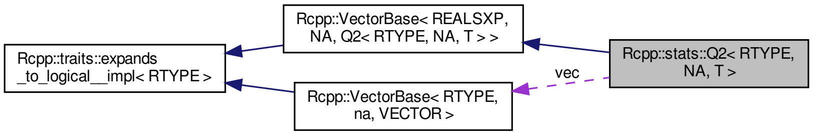
6.530 Rcpp::stats::Q2< RTYPE, NA, T > Class Template Reference

```
#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

6.530.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q2< RTYPE, NA, T >

Definition at line 268 of file dpq.h.

6.530.2 Member Typedef Documentation

6.530.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::Q2< RTYPE, NA, T >::FunPtr) (double, double, double, int, int)
```

Definition at line 271 of file dpq.h.

6.530.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q2< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 270 of file dpq.h.

6.530.3 Constructor & Destructor Documentation

6.530.3.1 Q2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q2< RTYPE, NA, T >::Q2 (
             FunPtr ptr_,
             const VEC_TYPE & vec_,
             double p0_,
             double p1_,
             bool lower_tail = true,
             bool log_ = false )  [inline]
```

Definition at line 273 of file dpq.h.

6.530.4 Member Function Documentation

6.530.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q2< RTYPE, NA, T >::operator[] (
             int i ) const  [inline]
```

Definition at line 277 of file dpq.h.

References Rcpp::stats::Q2< RTYPE, NA, T >::log, Rcpp::stats::Q2< RTYPE, NA, T >::lower, Rcpp::stats::Q2< RTYPE, NA, T >::p0, Rcpp::stats::Q2< RTYPE, NA, T >::p1, Rcpp::stats::Q2< RTYPE, NA, T >::ptr, and Rcpp::stats::Q2< RTYPE, NA, T >::vec.

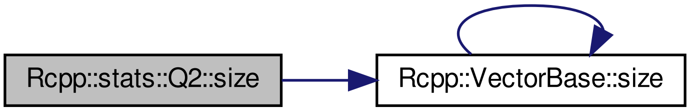
6.530.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 281 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q2< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.530.5 Member Data Documentation

6.530.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::log  [private]
```

Definition at line 287 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ]().

6.530.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::lower  [private]
```

Definition at line 287 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ]().

6.530.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q2< RTYPE, NA, T >::p0  [private]
```

Definition at line 286 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ]().

6.530.5.4 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q2< RTYPE, NA, T >::p1  [private]
```

Definition at line 286 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ]().

6.530.5.5 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::Q2< RTYPE, NA, T >::ptr  [private]
```

Definition at line 284 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ]().

6.530.5.6 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q2< RTYPE, NA, T >::vec  [private]
```

Definition at line 285 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::Q2< RTYPE, NA, T >::size().

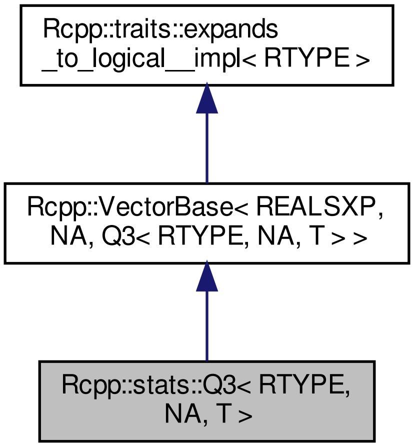
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

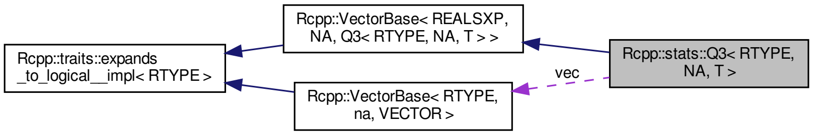
6.531 Rcpp::stats::Q3< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q3< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Q3< RTYPE, NA, T > >
Rcpp::stats::Q3< RTYPE, NA, T >

Collaboration diagram for Rcpp::stats::Q3< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Q3< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
vec
Rcpp::stats::Q3< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef double(∗ FunPtr) (double, double, double, double, int, int)

Public Member Functions

- Q3 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, double p1_, double p2_, bool lower_tail=true, bool log_=false)
- double operator[ ] (int i) const
- int size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec
- double p0
- double p1
- double p2
- int lower
- int log

6.531.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::stats::Q3< RTYPE, NA, T >

Definition at line 292 of file dpq.h.

6.531.2 Member Typedef Documentation

6.531.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::Q3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int,
int)
```

Definition at line 295 of file dpq.h.

6.531.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q3< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 294 of file dpq.h.

6.531.3 Constructor & Destructor Documentation

6.531.3.1 Q3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q3< RTYPE, NA, T >::Q3 (
            FunPtr ptr_,
            const VEC_TYPE & vec_,
            double p0_,
            double p1_,
            double p2_,
            bool lower_tail = true,
            bool log_ = false )  [inline]
```

Definition at line 297 of file dpq.h.

6.531.4 Member Function Documentation

6.531.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::operator[] (
            int i ) const  [inline]
```

Definition at line 301 of file dpq.h.

References Rcpp::stats::Q3< RTYPE, NA, T >::log, Rcpp::stats::Q3< RTYPE, NA, T >::lower, Rcpp::stats::Q3< RTYPE, NA, T >::p0, Rcpp::stats::Q3< RTYPE, NA, T >::p1, Rcpp::stats::Q3< RTYPE, NA, T >::p2, Rcpp::stats::Q3< RTYPE, NA, T >::ptr, and Rcpp::stats::Q3< RTYPE, NA, T >::vec.

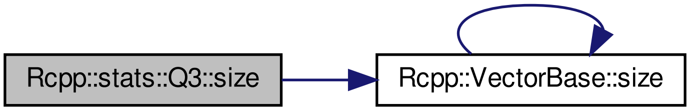
6.531.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q3< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 305 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q3< RTYPE, NA, T >::vec.

Here is the call graph for this function:



6.531.5 Member Data Documentation

6.531.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q3< RTYPE, NA, T >::log  [private]
```

Definition at line 311 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q3< RTYPE, NA, T >::lower  [private]
```

Definition at line 311 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.3 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::p0  [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.4 p1

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::p1  [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.5 p2

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::p2  [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.6 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::Q3< RTYPE, NA, T >::ptr  [private]
```

Definition at line 308 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ]().

6.531.5.7 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q3< RTYPE, NA, T >::vec  [private]
```

Definition at line 309 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::Q3< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

6.532 Rcpp::traits::expands_to_logical__impl< LGLSXP >::r_expands_to_logical Struct Reference

```
#include <expands_to_logical.h>
```

6.532.1 Detailed Description

Definition at line 37 of file expands_to_logical.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands_to_logical.h

6.533 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface Struct Reference

```
#include <MatrixBase.h>
```

6.533.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface

Definition at line 35 of file MatrixBase.h.

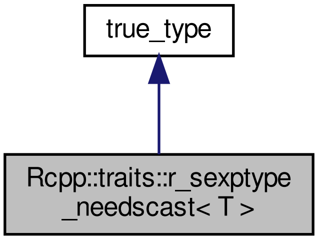
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

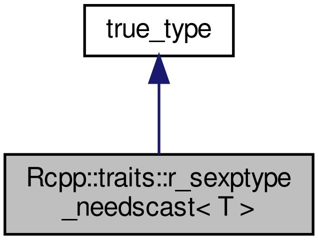
6.534 Rcpp::traits::r_sexptype_needscast< T > Struct Template Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< T >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< T >:



Additional Inherited Members

6.534.1 Detailed Description

template<typename T>
struct Rcpp::traits::r_sexptype_needscast< T >

Indicates if a primitive type needs a static_cast

Definition at line 69 of file r_sexptype_traits.h.

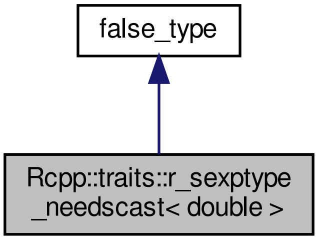
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

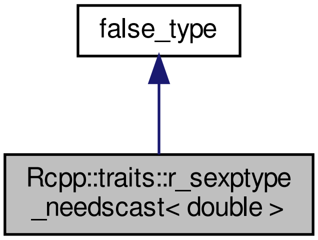
6.535 Rcpp::traits::r_sexptype_needscast< double > Struct Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< double >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< double >:



Additional Inherited Members

6.535.1 Detailed Description

Definition at line 71 of file r_sexptype_traits.h.

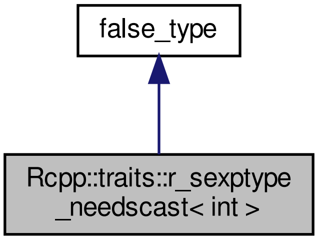
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

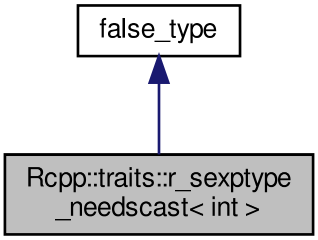
6.536 Rcpp::traits::r_sexptype_needscast< int > Struct Reference

```
#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

6.536.1 Detailed Description

Definition at line 70 of file r_sexptype_traits.h.

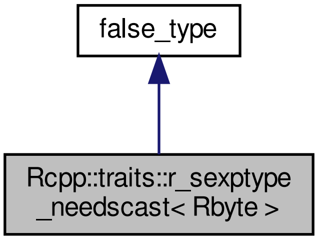
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

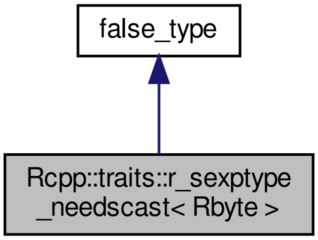
6.537 Rcpp::traits::r_sexptype_needscast< Rbyte > Struct Reference

```
#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

6.537.1 Detailed Description

Definition at line 73 of file r_sexptype_traits.h.

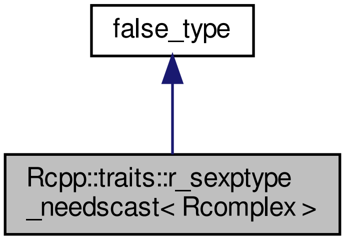
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

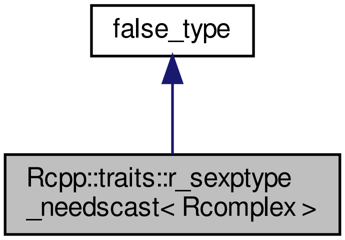
6.538 Rcpp::traits::r_sexptype_needscast< Rcomplex > Struct Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r_sexptype_needscast< Rcomplex >:



Collaboration diagram for Rcpp::traits::r_sexptype_needscast< Rcomplex >:



Additional Inherited Members

6.538.1 Detailed Description

Definition at line 72 of file r_sexptype_traits.h.

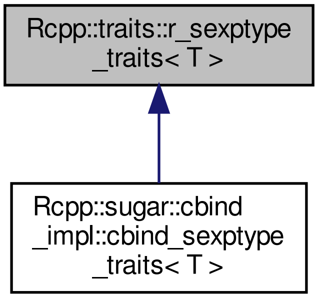
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.539 Rcpp::traits::r_sexptype_traits< T > Struct Template Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r_sexptype_traits< T >:



Public Types

- enum { rtype = VECSXP }

6.539.1 Detailed Description

template<typename T>
struct Rcpp::traits::r_sexptype_traits< T >

template that returns the SEXP type that is appropriate for the type T, this is allways VECSXP (lists) unless it is specialized

Definition at line 33 of file r_sexptype_traits.h.

6.539.2 Member Enumeration Documentation

6.539.2.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

rtype	

Definition at line 33 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.540 Rcpp::traits::r_sexptype_traits< bool > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = LGLSXP }

6.540.1 Detailed Description

Definition at line 38 of file r_sexptype_traits.h.

6.540.2 Member Enumeration Documentation

6.540.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 38 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.541 Rcpp::traits::r_sexptype_traits< const double > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.541.1 Detailed Description

Definition at line 37 of file r_sexptype_traits.h.

6.541.2 Member Enumeration Documentation

6.541.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 37 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.542 Rcpp::traits::r_sexptype_traits< const int > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = INTSXP }

6.542.1 Detailed Description

Definition at line 35 of file r_sexptype_traits.h.

6.542.2 Member Enumeration Documentation

6.542.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 35 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.543 Rcpp::traits::r_sexptype_traits< double > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.543.1 Detailed Description

Definition at line 36 of file r_sexptype_traits.h.

6.543.2 Member Enumeration Documentation

6.543.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 36 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.544 Rcpp::traits::r_sexptype_traits< float > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.544.1 Detailed Description

Definition at line 45 of file r_sexptype_traits.h.

6.544.2 Member Enumeration Documentation

6.544.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 45 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.545 Rcpp::traits::r_sexptype_traits< int > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = INTSXP }

6.545.1 Detailed Description

Definition at line 34 of file r_sexptype_traits.h.

6.545.2 Member Enumeration Documentation

6.545.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 34 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.546 Rcpp::traits::r_sexptype_traits< long > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.546.1 Detailed Description

Definition at line 49 of file r_sexptype_traits.h.

6.546.2 Member Enumeration Documentation

6.546.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 49 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.547 Rcpp::traits::r_sexptype_traits< long double > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.547.1 Detailed Description

Definition at line 55 of file r_sexptype_traits.h.

6.547.2 Member Enumeration Documentation

6.547.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 55 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.548 Rcpp::traits::r_sexptype_traits< Rbyte > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = RAWSXP }

6.548.1 Detailed Description

Definition at line 41 of file r_sexptype_traits.h.

6.548.2 Member Enumeration Documentation

6.548.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 41 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.549 Rcpp::traits::r_sexptype_traits< Rcomplex > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = CPLXSXP }

6.549.1 Detailed Description

Definition at line 40 of file r_sexptype_traits.h.

6.549.2 Member Enumeration Documentation

6.549.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 40 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.550 Rcpp::traits::r_sexptype_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

Public Types

- enum { rtype = REALSXP }

6.550.1 Detailed Description

Definition at line 37 of file Date.h.

6.550.2 Member Enumeration Documentation

6.550.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 38 of file Date.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Date.h

6.551 Rcpp::traits::r_sexptype_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

Public Types

- enum { rtype = REALSXP }

6.551.1 Detailed Description

Definition at line 37 of file Datetime.h.

6.551.2 Member Enumeration Documentation

6.551.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 38 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

6.552 Rcpp::traits::r_sexptype_traits< Rcpp::String > Struct Reference

```
#include <String.h>
```

Public Types

- enum { rtype = STRSXP }

6.552.1 Detailed Description

Definition at line 677 of file String.h.

6.552.2 Member Enumeration Documentation

6.552.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 677 of file String.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/String.h

6.553 Rcpp::traits::r_sexptype_traits< short > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = INTSXP }

6.553.1 Detailed Description

Definition at line 59 of file r_sexptype_traits.h.

6.553.2 Member Enumeration Documentation

6.553.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 59 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.554 Rcpp::traits::r_sexptype_traits< std::complex< double > > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = CPLXSXP }

6.554.1 Detailed Description

Definition at line 63 of file r_sexptype_traits.h.

6.554.2 Member Enumeration Documentation

6.554.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 63 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.555 Rcpp::traits::r_sexptype_traits< std::complex< float > > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = CPLXSXP }

6.555.1 Detailed Description

Definition at line 64 of file r_sexptype_traits.h.

6.555.2 Member Enumeration Documentation

6.555.2.1 anonymous enum

```
anonymous enum
```

Enumerator

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

6.556 Rcpp::traits::r_sexptype_traits< std::string > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = STRSXP }

6.556.1 Detailed Description

Definition at line 39 of file r_sexptype_traits.h.

6.556.2 Member Enumeration Documentation

6.556.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 39 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.557 Rcpp::traits::r_sexptype_traits< unsigned int > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.557.1 Detailed Description

Definition at line 44 of file r_sexptype_traits.h.

6.557.2 Member Enumeration Documentation

6.557.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 44 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.558 Rcpp::traits::r_sexptype_traits< unsigned long > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = REALSXP }

6.558.1 Detailed Description

Definition at line 50 of file r_sexptype_traits.h.

6.558.2 Member Enumeration Documentation

6.558.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 50 of file r_sexptype_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

6.559 Rcpp::traits::r_sexptype_traits< unsigned short > Struct Reference

```
#include <r_sexptype_traits.h>
```

Public Types

- enum { rtype = INTSXP }

6.559.1 Detailed Description

Definition at line 60 of file r_sexptype_traits.h.

6.559.2 Member Enumeration Documentation

6.559.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 60 of file r_sexptype_traits.h.

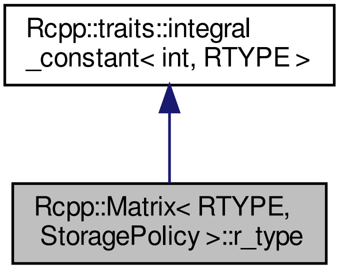
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_sexptype_traits.h

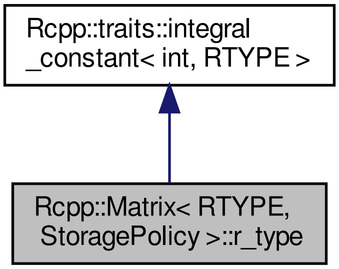
6.560 Rcpp::Matrix< RTYPE, StoragePolicy >::r_type Struct Reference

```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r_type:



Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r_type:



Additional Inherited Members

6.560.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::Matrix< RTYPE, StoragePolicy >::r_type

Definition at line 34 of file Matrix.h.

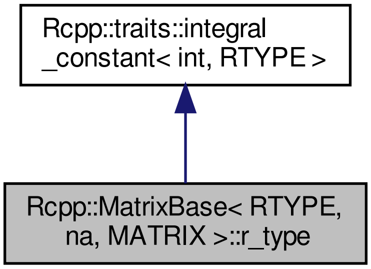
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/Matrix.h

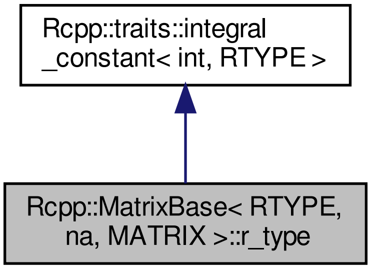
6.561 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type Struct Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type:


Rcpp::traits::integral_constant< int, RTYPE >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type

Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type:


Rcpp::traits::integral_constant< int, RTYPE >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type

Additional Inherited Members

6.561.1 Detailed Description

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type

Definition at line 34 of file MatrixBase.h.

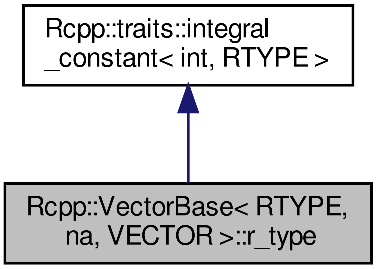
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

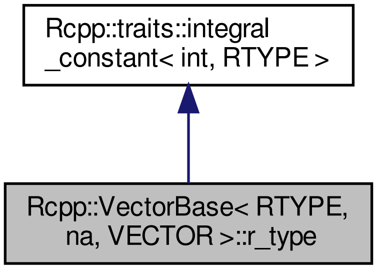
6.562 Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type Struct Reference

```
#include <VectorBase.h>
```

Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type:



Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type:



Additional Inherited Members

6.562.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type

Definition at line 32 of file VectorBase.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

6.563 Rcpp::traits::r_type_enum_tag Struct Reference

```
#include <r_type_traits.h>
```

6.563.1 Detailed Description

identifies an enum. conversions from/to int is used

Definition at line 97 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.564 Rcpp::traits::r_type_generic_tag Struct Reference

```
#include <r_type_traits.h>
```

6.564.1 Detailed Description

Default

Definition at line 44 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.565 Rcpp::traits::r_type_module_object_const_pointer_tag Struct Reference

```
#include <r_type_traits.h>
```

6.565.1 Detailed Description

identifies a module object const pointer

Definition at line 76 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.566 Rcpp::traits::r_type_module_object_const_reference_tag Struct Reference

```
#include <r_type_traits.h>
```

6.566.1 Detailed Description

identifies a const reference to a module object.

Definition at line 92 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.567 Rcpp::traits::r_type_module_object_pointer_tag Struct Reference

```
#include <r_type_traits.h>
```

6.567.1 Detailed Description

identifies a module object pointer

Definition at line 71 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.568 Rcpp::traits::r_type_module_object_reference_tag Struct Reference

```
#include <r_type_traits.h>
```

6.568.1 Detailed Description

identifies a reference to a module object.

Definition at line 87 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.569 Rcpp::traits::r_type_module_object_tag Struct Reference

```
#include <r_type_traits.h>
```

6.569.1 Detailed Description

identifies a module object. Implementers of modules can define the r_type_traits to show that their object is handled

Definition at line 82 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.570 Rcpp::traits::r_type_pair_tag Struct Reference

```
#include <r_type_traits.h>
```

6.570.1 Detailed Description

Identifies a pair<const KEY, VALUE>, used to wrap map<KEY, VALUE> VALUE can be anything that wraps, KEY can be anything we can build a String from

Definition at line 66 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.571 Rcpp::traits::r_type_pairstring_generic_tag Struct Reference

```
#include <r_type_traits.h>
```

6.571.1 Detailed Description

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

6.572 Rcpp::traits::r_type_pairstring_primitive_tag Struct Reference

```
#include <r_type_traits.h>
```

6.572.1 Detailed Description

Identifies that the type if pair<const std::string,T> where T is a primitive type

Definition at line 50 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.573 Rcpp::traits::r_type_pairstring_string_tag Struct Reference

```
#include <r_type_traits.h>
```

6.573.1 Detailed Description

Identifies that the associated type is pair<const std::(w)?string,std::(w)?string>

Definition at line 55 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.574 Rcpp::traits::r_type_primitive_tag Struct Reference

```
#include <r_type_traits.h>
```

6.574.1 Detailed Description

Identifies a primitive type that needs no special handling int, double, Rbyte, Rcomplex

Definition at line 33 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.575 Rcpp::traits::r_type_RcppString_tag Struct Reference

```
#include <r_type_traits.h>
```

6.575.1 Detailed Description

Definition at line 99 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.576 Rcpp::traits::r_type_string_tag Struct Reference

```
#include <r_type_traits.h>
```

6.576.1 Detailed Description

Identifies that the associated type can be 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

6.577 Rcpp::traits::r_type_traits< T > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_generic_tag r_category

6.577.1 Detailed Description

template<typename T>
struct Rcpp::traits::r_type_traits< T >

R type trait. Helps wrap.

Definition at line 104 of file r_type_traits.h.

6.577.2 Member Typedef Documentation

6.577.2.1 r_category

```
template<typename T >
typedef r_type_generic_tag Rcpp::traits::r_type_traits< T >::r_category
```

Definition at line 104 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.578 Rcpp::traits::r_type_traits< bool > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.578.1 Detailed Description

Definition at line 141 of file r_type_traits.h.

6.578.2 Member Typedef Documentation

6.578.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< bool >::r_category
```

Definition at line 141 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.579 Rcpp::traits::r_type_traits< char > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.579.1 Detailed Description

Definition at line 144 of file r_type_traits.h.

6.579.2 Member Typedef Documentation

6.579.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< char >::r_category
```

Definition at line 144 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.580 Rcpp::traits::r_type_traits< const char * > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.580.1 Detailed Description

Definition at line 149 of file r_type_traits.h.

6.580.2 Member Typedef Documentation

6.580.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< const char * >::r_category
```

Definition at line 149 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.581 Rcpp::traits::r_type_traits< const double > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.581.1 Detailed Description

Definition at line 138 of file r_type_traits.h.

6.581.2 Member Typedef Documentation

6.581.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< const double >::r_category
```

Definition at line 138 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.582 Rcpp::traits::r_type_traits< const int > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.582.1 Detailed Description

Definition at line 136 of file r_type_traits.h.

6.582.2 Member Typedef Documentation

6.582.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< const int >::r_category
```

Definition at line 136 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.583 Rcpp::traits::r_type_traits< const wchar_t * > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.583.1 Detailed Description

Definition at line 150 of file r_type_traits.h.

6.583.2 Member Typedef Documentation

6.583.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< const wchar_t * >::r_category
```

Definition at line 150 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.584 Rcpp::traits::r_type_traits< double > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.584.1 Detailed Description

Definition at line 137 of file r_type_traits.h.

6.584.2 Member Typedef Documentation

6.584.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< double >::r_category
```

Definition at line 137 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.585 Rcpp::traits::r_type_traits< float > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.585.1 Detailed Description

Definition at line 148 of file r_type_traits.h.

6.585.2 Member Typedef Documentation

6.585.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< float >::r_category
```

Definition at line 148 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.586 Rcpp::traits::r_type_traits< int > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.586.1 Detailed Description

Definition at line 135 of file r_type_traits.h.

6.586.2 Member Typedef Documentation

6.586.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< int >::r_category
```

Definition at line 135 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.587 Rcpp::traits::r_type_traits< long > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.587.1 Detailed Description

Definition at line 153 of file r_type_traits.h.

6.587.2 Member Typedef Documentation

6.587.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< long >::r_category
```

Definition at line 153 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.588 Rcpp::traits::r_type_traits< long double > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.588.1 Detailed Description

Definition at line 161 of file r_type_traits.h.

6.588.2 Member Typedef Documentation

6.588.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< long double >::r_category
```

Definition at line 161 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.589 Rcpp::traits::r_type_traits< Rbyte > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.589.1 Detailed Description

Definition at line 139 of file r_type_traits.h.

6.589.2 Member Typedef Documentation

6.589.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rbyte >::r_category
```

Definition at line 139 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.590 Rcpp::traits::r_type_traits< Rcomplex > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.590.1 Detailed Description

Definition at line 140 of file r_type_traits.h.

6.590.2 Member Typedef Documentation

6.590.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rcomplex >::r_category
```

Definition at line 140 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.591 Rcpp::traits::r_type_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.591.1 Detailed Description

Definition at line 31 of file Date.h.

6.591.2 Member Typedef Documentation

6.591.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rcpp::Date >::r_category
```

Definition at line 32 of file Date.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Date.h

6.592 Rcpp::traits::r_type_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.592.1 Detailed Description

Definition at line 31 of file Datetime.h.

6.592.2 Member Typedef Documentation

6.592.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rcpp::Datetime >::r_category
```

Definition at line 32 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

6.593 Rcpp::traits::r_type_traits< Rcpp::object< T > > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_module_object_pointer_tag r_category

6.593.1 Detailed Description

template<typename T>
struct Rcpp::traits::r_type_traits< Rcpp::object< T > >

module object type

Definition at line 109 of file r_type_traits.h.

6.593.2 Member Typedef Documentation

6.593.2.1 r_category

```
template<typename T >
typedef r_type_module_object_pointer_tag Rcpp::traits::r_type_traits< Rcpp::object< T > >↩
::r_category
```

Definition at line 109 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.594 Rcpp::traits::r_type_traits< Rcpp::String > Struct Reference

```
#include <String.h>
```

Public Types

- typedef r_type_RcppString_tag r_category

6.594.1 Detailed Description

Definition at line 676 of file String.h.

6.594.2 Member Typedef Documentation

6.594.2.1 r_category

```
typedef r_type_RcppString_tag Rcpp::traits::r_type_traits< Rcpp::String >::r_category
```

Definition at line 676 of file String.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/String.h

6.595 Rcpp::traits::r_type_traits< short > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.595.1 Detailed Description

Definition at line 165 of file r_type_traits.h.

6.595.2 Member Typedef Documentation

6.595.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< short >::r_category
```

Definition at line 165 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.596 Rcpp::traits::r_type_traits< std::complex< double > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.596.1 Detailed Description

Definition at line 173 of file r_type_traits.h.

6.596.2 Member Typedef Documentation

6.596.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::complex< double > >::r_category
```

Definition at line 173 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.597 Rcpp::traits::r_type_traits< std::complex< float > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.597.1 Detailed Description

Definition at line 177 of file r_type_traits.h.

6.597.2 Member Typedef Documentation

6.597.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::complex< float > >::r_category
```

Definition at line 177 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.598 Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pair_tag r_category

6.598.1 Detailed Description

template<typename KEY, typename VALUE>
struct Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >

Definition at line 113 of file r_type_traits.h.

6.598.2 Member Typedef Documentation

6.598.2.1 r_category

```
template<typename KEY , typename VALUE >
typedef r_type_pair_tag Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >::r_category
```

Definition at line 114 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.599 Rcpp::traits::r_type_traits< std::pair< const std::string, bool > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.599.1 Detailed Description

Definition at line 126 of file r_type_traits.h.

6.599.2 Member Typedef Documentation

6.599.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
bool > >::r_category
```

Definition at line 126 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.600 Rcpp::traits::r_type_traits< std::pair< const std::string, char > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_string_tag r_category

6.600.1 Detailed Description

Definition at line 129 of file r_type_traits.h.

6.600.2 Member Typedef Documentation

6.600.2.1 r_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
char > >::r_category
```

Definition at line 129 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.601 Rcpp::traits::r_type_traits< std::pair< const std::string, const int > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.601.1 Detailed Description

Definition at line 122 of file r_type_traits.h.

6.601.2 Member Typedef Documentation

6.601.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
const int > >::r_category
```

Definition at line 122 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.602 Rcpp::traits::r_type_traits< std::pair< const std::string, double > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.602.1 Detailed Description

Definition at line 123 of file r_type_traits.h.

6.602.2 Member Typedef Documentation

6.602.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
double > >::r_category
```

Definition at line 123 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.603 Rcpp::traits::r_type_traits< std::pair< const std::string, float > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.603.1 Detailed Description

Definition at line 133 of file r_type_traits.h.

6.603.2 Member Typedef Documentation

6.603.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
float > >::r_category
```

Definition at line 133 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.604 Rcpp::traits::r_type_traits< std::pair< const std::string, int > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.604.1 Detailed Description

Definition at line 121 of file r_type_traits.h.

6.604.2 Member Typedef Documentation

6.604.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
int > >::r_category
```

Definition at line 121 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.605 Rcpp::traits::r_type_traits< std::pair< const std::string, long > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.605.1 Detailed Description

Definition at line 154 of file r_type_traits.h.

6.605.2 Member Typedef Documentation

6.605.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, long >
>::r_category
```

Definition at line 154 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.606 Rcpp::traits::r_type_traits< std::pair< const std::string, long double > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.606.1 Detailed Description

Definition at line 162 of file r_type_traits.h.

6.606.2 Member Typedef Documentation

6.606.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, long
double > >::r_category
```

Definition at line 162 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.607 Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.607.1 Detailed Description

Definition at line 124 of file r_type_traits.h.

6.607.2 Member Typedef Documentation

6.607.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
Rbyte > >::r_category
```

Definition at line 124 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.608 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.608.1 Detailed Description

Definition at line 125 of file r_type_traits.h.

6.608.2 Member Typedef Documentation

6.608.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
Rcomplex > >::r_category
```

Definition at line 125 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.609 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > > Struct Reference

```
#include <Date.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.609.1 Detailed Description

Definition at line 34 of file Date.h.

6.609.2 Member Typedef Documentation

6.609.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date
> >::r_category
```

Definition at line 35 of file Date.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Date.h

6.610 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > > Struct Reference

```
#include <Datetime.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.610.1 Detailed Description

Definition at line 34 of file Datetime.h.

6.610.2 Member Typedef Documentation

6.610.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime
> >::r_category
```

Definition at line 35 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

6.611 Rcpp::traits::r_type_traits< std::pair< const std::string, short > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.611.1 Detailed Description

Definition at line 166 of file r_type_traits.h.

6.611.2 Member Typedef Documentation

6.611.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, short >
>::r_category
```

Definition at line 166 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.612 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.612.1 Detailed Description

Definition at line 174 of file r_type_traits.h.

6.612.2 Member Typedef Documentation

6.612.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, std←
::complex< double > > >::r_category
```

Definition at line 174 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.613 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.613.1 Detailed Description

Definition at line 178 of file r_type_traits.h.

6.613.2 Member Typedef Documentation

6.613.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, std↩
::complex< float > > >::r_category
```

Definition at line 178 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.614 Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_string_tag r_category

6.614.1 Detailed Description

Definition at line 127 of file r_type_traits.h.

6.614.2 Member Typedef Documentation

6.614.2.1 r_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
std::string > >::r_category
```

Definition at line 127 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.615 Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_string_tag r_category

6.615.1 Detailed Description

Definition at line 128 of file r_type_traits.h.

6.615.2 Member Typedef Documentation

6.615.2.1 r_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
std::wstring > >::r_category
```

Definition at line 128 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.616 Rcpp::traits::r_type_traits< std::pair< const std::string, T > > Struct Template Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_generic_tag r_category

6.616.1 Detailed Description

template<typename T>
struct Rcpp::traits::r_type_traits< std::pair< const std::string, T > >

special cases pair<string,T> to deal with map<string,T> etc ...

Definition at line 120 of file r_type_traits.h.

6.616.2 Member Typedef Documentation

6.616.2.1 r_category

```
template<typename T >
typedef r_type_pairstring_generic_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
T > >::r_category
```

Definition at line 120 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.617 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned int > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_primitive_tag r_category

6.617.1 Detailed Description

Definition at line 132 of file r_type_traits.h.

6.617.2 Member Typedef Documentation

6.617.2.1 r_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
unsigned int > >::r_category
```

Definition at line 132 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.618 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned long > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.618.1 Detailed Description

Definition at line 158 of file r_type_traits.h.

6.618.2 Member Typedef Documentation

6.618.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned
long > >::r_category
```

Definition at line 158 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.619 Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned short > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.619.1 Detailed Description

Definition at line 170 of file r_type_traits.h.

6.619.2 Member Typedef Documentation

6.619.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned
short > >::r_category
```

Definition at line 170 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.620 Rcpp::traits::r_type_traits< std::pair< const std::string, wchar_t > > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_pairstring_string_tag r_category

6.620.1 Detailed Description

Definition at line 130 of file r_type_traits.h.

6.620.2 Member Typedef Documentation

6.620.2.1 r_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,
wchar_t > >::r_category
```

Definition at line 130 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.621 Rcpp::traits::r_type_traits< std::string > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.621.1 Detailed Description

Definition at line 142 of file r_type_traits.h.

6.621.2 Member Typedef Documentation

6.621.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< std::string >::r_category
```

Definition at line 142 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.622 Rcpp::traits::r_type_traits< std::wstring > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.622.1 Detailed Description

Definition at line 143 of file r_type_traits.h.

6.622.2 Member Typedef Documentation

6.622.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< std::wstring >::r_category
```

Definition at line 143 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.623 Rcpp::traits::r_type_traits< unsigned int > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.623.1 Detailed Description

Definition at line 147 of file r_type_traits.h.

6.623.2 Member Typedef Documentation

6.623.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< unsigned int >::r_category
```

Definition at line 147 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.624 Rcpp::traits::r_type_traits< unsigned long > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.624.1 Detailed Description

Definition at line 157 of file r_type_traits.h.

6.624.2 Member Typedef Documentation

6.624.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< unsigned long >::r_category
```

Definition at line 157 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.625 Rcpp::traits::r_type_traits< unsigned short > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_primitive_tag r_category

6.625.1 Detailed Description

Definition at line 169 of file r_type_traits.h.

6.625.2 Member Typedef Documentation

6.625.2.1 r_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< unsigned short >::r_category
```

Definition at line 169 of file r_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.626 Rcpp::traits::r_type_traits< wchar_t > Struct Reference

```
#include <r_type_traits.h>
```

Public Types

- typedef r_type_string_tag r_category

6.626.1 Detailed Description

Definition at line 145 of file r_type_traits.h.

6.626.2 Member Typedef Documentation

6.626.2.1 r_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< wchar_t >::r_category
```

Definition at line 145 of file r_type_traits.h.

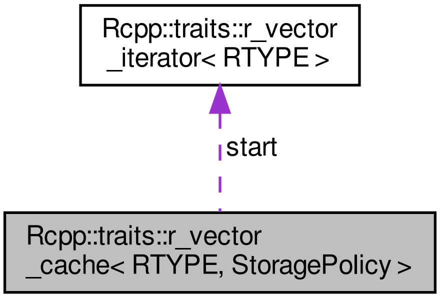
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/r_type_traits.h

6.627 Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy > Class Template Reference

```
#include <traits.h>
```

Collaboration diagram for Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >:



Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef r_vector_iterator< RTYPE >::type iterator
- typedef r_vector_const_iterator< RTYPE >::type const_iterator
- typedef r_vector_proxy< RTYPE >::type proxy
- typedef r_vector_const_proxy< RTYPE >::type const_proxy
- typedef storage_type< RTYPE >::type storage_type

Public Member Functions

- r_vector_cache ()
- void update (const VECTOR &v)
- iterator get () const
- const_iterator get_const () const
- proxy ref ()
- proxy ref (R_xlen_t i)
- proxy ref () const
- proxy ref (R_xlen_t i) const

Private Attributes

- iterator start

6.627.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >

Definition at line 29 of file traits.h.

6.627.2 Member Typedef Documentation

6.627.2.1 const_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_const_iterator<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy
>::const_iterator
```

Definition at line 33 of file traits.h.

6.627.2.2 const_proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_const_proxy<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >↩
::const_proxy
```

Definition at line 35 of file traits.h.

6.627.2.3 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_iterator<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >↩
::iterator
```

Definition at line 32 of file traits.h.

6.627.2.4 proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_proxy<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::proxy
```

Definition at line 34 of file traits.h.

6.627.2.5 storage_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef storage_type<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::storage_type
```

Definition at line 36 of file traits.h.

6.627.2.6 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy
>::VECTOR
```

Definition at line 31 of file traits.h.

6.627.3 Constructor & Destructor Documentation

6.627.3.1 r_vector_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::r_vector_cache ( )  [inline]
```

Definition at line 38 of file traits.h.

6.627.4 Member Function Documentation

6.627.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get ( ) const  [inline]
```

Definition at line 42 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.2 get_const()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get_const ( ) const  [inline]
```

Definition at line 43 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.3 ref() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref ( )  [inline]
```

Definition at line 45 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.4 ref() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref ( ) const  [inline]
```

Definition at line 48 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.5 ref() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref (
             R_xlen_t i )  [inline]
```

Definition at line 46 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.6 ref() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref (
             R_xlen_t i ) const  [inline]
```

Definition at line 49 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.4.7 update()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::update (
             const VECTOR & v )  [inline]
```

Definition at line 39 of file traits.h.

References Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start.

6.627.5 Member Data Documentation

6.627.5.1 start

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start  [private]
```

Definition at line 52 of file traits.h.

Referenced by Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get(), Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get_const(), Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref(), and Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::update().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/traits.h

6.628 Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

Public Types

- typedef r_vector_cache< RTYPE, StoragePolicy > type

6.628.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >

Definition at line 84 of file traits.h.

6.628.2 Member Typedef Documentation

6.628.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_cache<RTYPE, StoragePolicy> Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >::type
```

Definition at line 85 of file traits.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/traits.h

6.629 Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

Public Types

- typedef proxy_cache< EXPRSXP, StoragePolicy > type

6.629.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy >

Definition at line 94 of file traits.h.

6.629.2 Member Typedef Documentation

6.629.2.1 type

```
template<template< class > class StoragePolicy>
typedef proxy_cache<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< EXPRSXP, Storage↩
Policy >::type
```

Definition at line 95 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/traits.h

6.630 Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

Public Types

- typedef proxy_cache< STRSXP, StoragePolicy > type

6.630.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >

Definition at line 98 of file traits.h.

6.630.2 Member Typedef Documentation

6.630.2.1 type

```
template<template< class > class StoragePolicy>
typedef proxy_cache<STRSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< STRSXP, Storage↩
Policy >::type
```

Definition at line 99 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/traits.h

6.631 Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

Public Types

- typedef proxy_cache< VECSXP, StoragePolicy > type

6.631.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >

Definition at line 90 of file traits.h.

6.631.2 Member Typedef Documentation

6.631.2.1 type

```
template<template< class > class StoragePolicy>
typedef proxy_cache<VECSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< VECSXP, Storage↩
Policy >::type
```

Definition at line 91 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/traits.h

6.632 Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef const storage_type< RTYPE >::type ∗ type

6.632.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy >

Definition at line 254 of file proxy.h.

6.632.2 Member Typedef Documentation

6.632.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const storage_type<RTYPE>::type* Rcpp::traits::r_vector_const_iterator< RTYPE, Storage↩
Policy >::type
```

Definition at line 255 of file proxy.h.

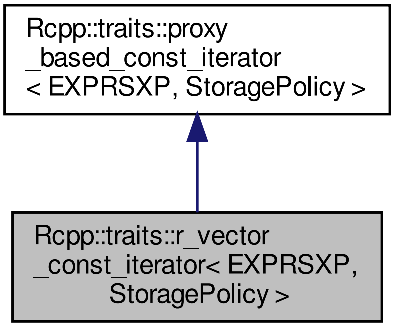
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

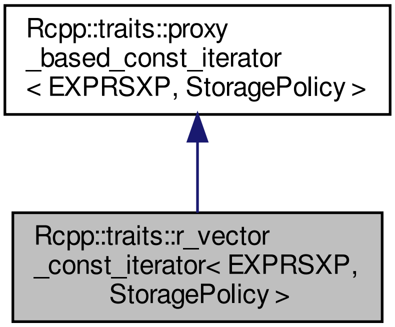
6.633 Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy >:



Additional Inherited Members

6.633.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy >

Definition at line 271 of file proxy.h.

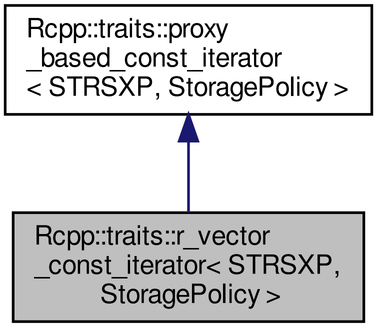
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

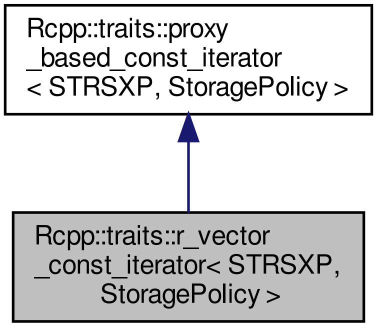
6.634 Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >:


Rcpp::traits::proxy _based_const_iterator < STRSXP, StoragePolicy >
Rcpp::traits::r_vector _const_iterator< STRSXP, StoragePolicy >

Collaboration diagram for Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >:


Rcpp::traits::proxy _based_const_iterator < STRSXP, StoragePolicy >
Rcpp::traits::r_vector _const_iterator< STRSXP, StoragePolicy >

Additional Inherited Members

6.634.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >

Definition at line 272 of file proxy.h.

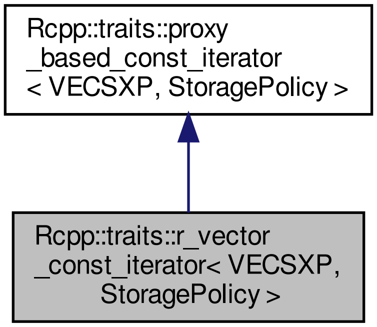
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

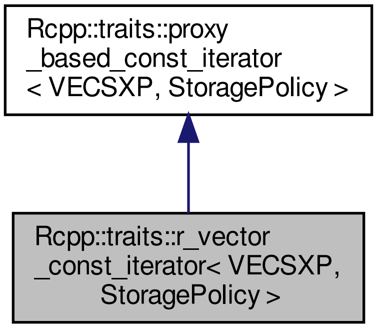
6.635 Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy > Struct Template Reference

`#include <proxy.h>`

Inheritance diagram for Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >:


Rcpp::traits::proxy _based_const_iterator < VECSXP, StoragePolicy >
Rcpp::traits::r_vector _const_iterator< VECSXP, StoragePolicy >

Collaboration diagram for Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >:


Rcpp::traits::proxy _based_const_iterator < VECSXP, StoragePolicy >
Rcpp::traits::r_vector _const_iterator< VECSXP, StoragePolicy >

Additional Inherited Members

6.635.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >

Definition at line 270 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.636 Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef const storage_type< RTYPE >::type & type

6.636.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy >

Definition at line 233 of file proxy.h.

6.636.2 Member Typedef Documentation

6.636.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const storage_type<RTYPE>::type& Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy
>::type
```

Definition at line 234 of file proxy.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.637 Rcpp::traits::r_vector_const_proxy< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::const_generic_proxy< EXPRSXP, StoragePolicy > type

6.637.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< EXPRSXP, StoragePolicy >

Definition at line 245 of file proxy.h.

6.637.2 Member Typedef Documentation

6.637.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::const_generic_proxy<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_const_proxy<
EXPRSXP, StoragePolicy >::type
```

Definition at line 246 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.638 Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::const_string_proxy< STRSXP, StoragePolicy > type

6.638.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy >

Definition at line 237 of file proxy.h.

6.638.2 Member Typedef Documentation

6.638.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::const_string_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_const_proxy<
STRSXP, StoragePolicy >::type
```

Definition at line 238 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.639 Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::const_generic_proxy< VECSXP, StoragePolicy > type

6.639.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy >

Definition at line 241 of file proxy.h.

6.639.2 Member Typedef Documentation

6.639.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::const_generic_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_const_proxy<
VECSXP, StoragePolicy >::type
```

Definition at line 242 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.640 Rcpp::traits::r_vector_element_converter< RTYPE > Struct Template Reference

```
#include <converter.h>
```

Public Types

- typedef ::Rcpp::internal::element_converter< RTYPE > type

6.640.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::r_vector_element_converter< RTYPE >

Definition at line 94 of file converter.h.

6.640.2 Member Typedef Documentation

6.640.2.1 type

```
template<int RTYPE>
typedef ::Rcpp::internal::element_converter<RTYPE> Rcpp::traits::r_vector_element_converter<
RTYPE >::type
```

Definition at line 95 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.641 Rcpp::traits::r_vector_element_converter< EXPRSXP > Struct Reference

```
#include <converter.h>
```

Public Types

- typedef ::Rcpp::internal::generic_element_converter< EXPRSXP > type

6.641.1 Detailed Description

Definition at line 103 of file converter.h.

6.641.2 Member Typedef Documentation

6.641.2.1 type

```
typedef ::Rcpp::internal::generic_element_converter<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

6.642 Rcpp::traits::r_vector_element_converter< STRSXP > Struct Reference

```
#include <converter.h>
```

Public Types

- typedef ::Rcpp::internal::string_element_converter< STRSXP > type

6.642.1 Detailed Description

Definition at line 97 of file converter.h.

6.642.2 Member Typedef Documentation

6.642.2.1 type

```
typedef ::Rcpp::internal::string_element_converter<STRSXP> Rcpp::traits::r_vector_element_converter<
STRSXP >::type
```

Definition at line 98 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.643 Rcpp::traits::r_vector_element_converter< VECSXP > Struct Reference

```
#include <converter.h>
```

Public Types

- typedef ::Rcpp::internal::generic_element_converter< VECSXP > type

6.643.1 Detailed Description

Definition at line 100 of file converter.h.

6.643.2 Member Typedef Documentation

6.643.2.1 type

```
typedef ::Rcpp::internal::generic_element_converter<VECSXP> Rcpp::traits::r_vector_element_converter<
VECSXP >::type
```

Definition at line 101 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/converter.h

6.644 Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef storage_type< RTYPE >::type ∗ type

6.644.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >

Definition at line 250 of file proxy.h.

6.644.2 Member Typedef Documentation

6.644.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef storage_type<RTYPE>::type* Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 251 of file proxy.h.

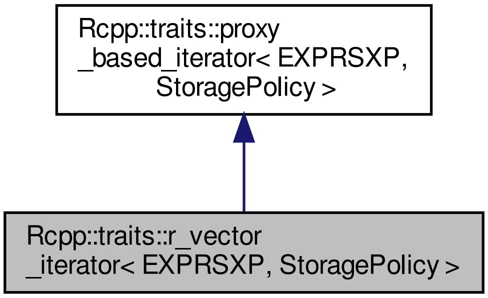
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

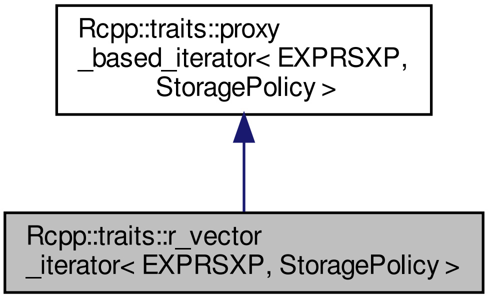
6.645 Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >:


Rcpp::traits::proxy _based_iterator< EXPRSXP, StoragePolicy >
Rcpp::traits::r_vector _iterator< EXPRSXP, StoragePolicy >

Collaboration diagram for Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >:


Rcpp::traits::proxy _based_iterator< EXPRSXP, StoragePolicy >
Rcpp::traits::r_vector _iterator< EXPRSXP, StoragePolicy >

Additional Inherited Members

6.645.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >

Definition at line 263 of file proxy.h.

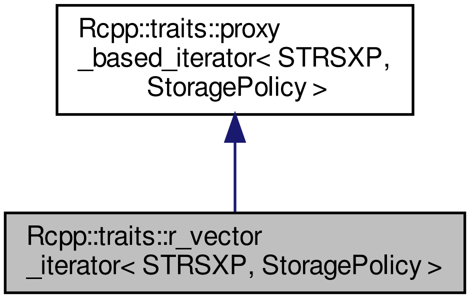
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

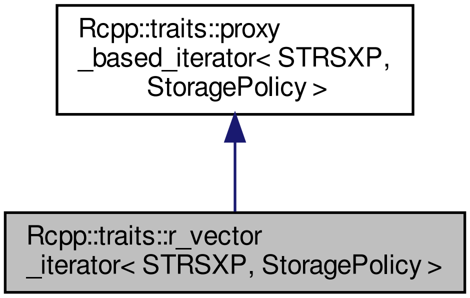
6.646 Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >:



Additional Inherited Members

6.646.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >

Definition at line 264 of file proxy.h.

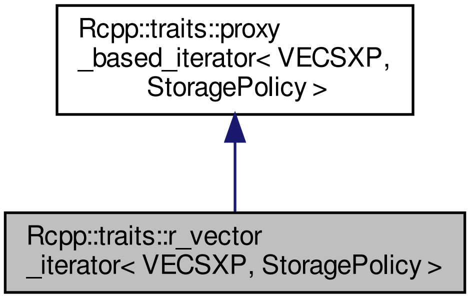
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

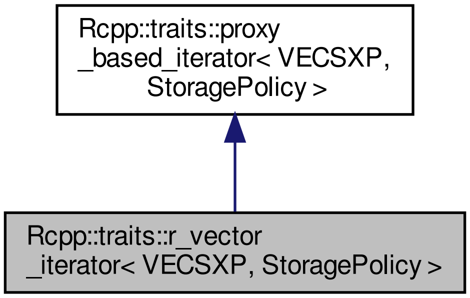
6.647 Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >:



Additional Inherited Members

6.647.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >

Definition at line 262 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.648 Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy > type

6.648.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy >

Definition at line 199 of file proxy.h.

6.648.2 Member Typedef Documentation

6.648.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::simple_name_proxy<RTYPE, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
RTYPE, StoragePolicy >::type
```

Definition at line 200 of file proxy.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.649 Rcpp::traits::r_vector_name_proxy< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::generic_name_proxy< EXPRSXP, StoragePolicy > type

6.649.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< EXPRSXP, StoragePolicy >

Definition at line 211 of file proxy.h.

6.649.2 Member Typedef Documentation

6.649.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_name_proxy<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
EXPRSXP, StoragePolicy >::type
```

Definition at line 212 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.650 Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::string_name_proxy< STRSXP, StoragePolicy > type

6.650.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy >

Definition at line 203 of file proxy.h.

6.650.2 Member Typedef Documentation

6.650.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::string_name_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
STRSXP, StoragePolicy >::type
```

Definition at line 204 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.651 Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::generic_name_proxy< VECSXP, StoragePolicy > type

6.651.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy >

Definition at line 207 of file proxy.h.

6.651.2 Member Typedef Documentation

6.651.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_name_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
VECSXP, StoragePolicy >::type
```

Definition at line 208 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.652 Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef storage_type< RTYPE >::type & type

6.652.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >

Definition at line 216 of file proxy.h.

6.652.2 Member Typedef Documentation

6.652.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef storage_type<RTYPE>::type& Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >::type
```

Definition at line 217 of file proxy.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.653 Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::generic_proxy< EXPRSXP, StoragePolicy > type

6.653.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy >

Definition at line 224 of file proxy.h.

6.653.2 Member Typedef Documentation

6.653.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_proxy<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_proxy<
EXPRSXP, StoragePolicy >::type
```

Definition at line 225 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.654 Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::string_proxy< STRSXP, StoragePolicy > type

6.654.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy >

Definition at line 220 of file proxy.h.

6.654.2 Member Typedef Documentation

6.654.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::string_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_proxy< STRSXP,
StoragePolicy >::type
```

Definition at line 221 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

6.655 Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Public Types

- typedef ::Rcpp::internal::generic_proxy< VECSXP, StoragePolicy > type

6.655.1 Detailed Description

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy >

Definition at line 228 of file proxy.h.

6.655.2 Member Typedef Documentation

6.655.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_proxy<
VECSXP, StoragePolicy >::type
```

Definition at line 229 of file proxy.h.

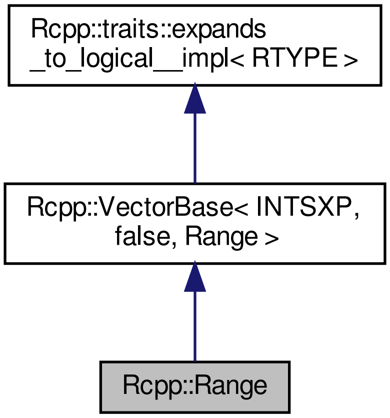
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

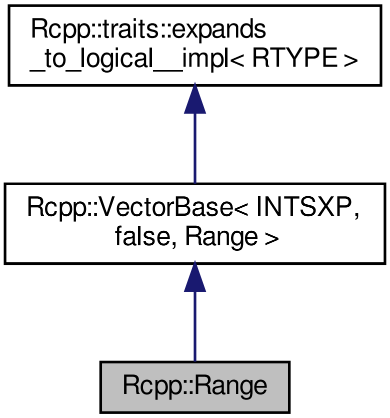
6.656 Rcpp::Range Class Reference

```
#include <Range.h>
```

Inheritance diagram for Rcpp::Range:



Collaboration diagram for Rcpp::Range:



Public Member Functions

- Range (R_xlen_t start_, R_xlen_t end__)
- R_xlen_t size () const
- R_xlen_t operator[ ] (R_xlen_t i) const
- Range & operator++ ()
- Range operator++ (int)
- Range & operator-- ()
- Range operator-- (int)
- Range & operator+= (int n)
- Range & operator-= (int n)
- Range operator+ (int n)
- Range operator- (int n)
- R_xlen_t get_start () const
- R_xlen_t get_end () const

Private Attributes

- R_xlen_t start
- R_xlen_t end_

Additional Inherited Members

6.656.1 Detailed Description

Definition at line 27 of file Range.h.

6.656.2 Constructor & Destructor Documentation

6.656.2.1 Range()

```
Rcpp::Range::Range (
            R_xlen_t start_,
            R_xlen_t end__ )  [inline]
```

Definition at line 29 of file Range.h.

Referenced by operator+(), and operator-().

6.656.3 Member Function Documentation

6.656.3.1 get_end()

```
R_xlen_t Rcpp::Range::get_end ( ) const  [inline]
```

Definition at line 83 of file Range.h.

References end_.

6.656.3.2 get_start()

```
R_xlen_t Rcpp::Range::get_start ( ) const  [inline]
```

Definition at line 81 of file Range.h.

References start.

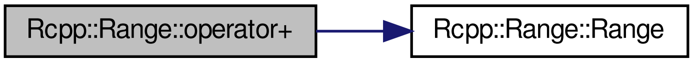
6.656.3.3 operator+()

```
Range Rcpp::Range::operator+ (
             int n )  [inline]
```

Definition at line 73 of file Range.h.

References end_, Range(), and start.

Here is the call graph for this function:



6.656.3.4 operator++() [1/2]

```
Range& Rcpp::Range::operator++ ( )  [inline]
```

Definition at line 43 of file Range.h.

References end_, and start.

6.656.3.5 operator++() [2/2]

```
Range Rcpp::Range::operator++ (
             int  )  [inline]
```

Definition at line 47 of file Range.h.

6.656.3.6 operator+=()

```
Range& Rcpp::Range::operator+= (
             int n )  [inline]
```

Definition at line 63 of file Range.h.

References end_, and start.

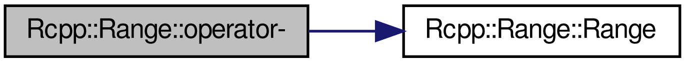
6.656.3.7 operator-()

```
Range Rcpp::Range::operator- (
             int n )  [inline]
```

Definition at line 77 of file Range.h.

References end_, Range(), and start.

Here is the call graph for this function:



6.656.3.8 operator--() [1/2]

```
Range& Rcpp::Range::operator-- ( )  [inline]
```

Definition at line 53 of file Range.h.

References end_, and start.

6.656.3.9 operator--() [2/2]

```
Range Rcpp::Range::operator-- (
             int  )  [inline]
```

Definition at line 57 of file Range.h.

6.656.3.10 operator-=()

```
Range& Rcpp::Range::operator-= (
             int n )  [inline]
```

Definition at line 68 of file Range.h.

References end_, and start.

6.656.3.11 operator[]()

```
R_xlen_t Rcpp::Range::operator[ ] (
            R_xlen_t i ) const  [inline]
```

Definition at line 39 of file Range.h.

References start.

6.656.3.12 size()

```
R_xlen_t Rcpp::Range::size ( ) const  [inline]
```

Definition at line 35 of file Range.h.

References end_, and start.

6.656.4 Member Data Documentation

6.656.4.1 end_

```
R_xlen_t Rcpp::Range::end_  [private]
```

Definition at line 87 of file Range.h.

Referenced by get_end(), operator+(), operator++(), operator+=(), operator-(), operator--(), operator-=(), and size().

6.656.4.2 start

```
R_xlen_t Rcpp::Range::start  [private]
```

Definition at line 86 of file Range.h.

Referenced by get_start(), operator+(), operator++(), operator+=(), operator-(), operator--(), operator-=(), operator[ ](), and size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/Range.h

6.657 Rcpp::sugar::Range< RTYPE, NA, T > Class Template Reference

```
#include <range.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Range (const T &obj_)
- operator Vector< RTYPE > ()

Private Attributes

- const T & obj
- STORAGE min_
- STORAGE max_
- STORAGE current

6.657.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Range< RTYPE, NA, T >

Definition at line 29 of file range.h.

6.657.2 Member Typedef Documentation

6.657.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Range< RTYPE, NA, T >::STORAGE
```

Definition at line 31 of file range.h.

6.657.3 Constructor & Destructor Documentation

6.657.3.1 Range()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Range< RTYPE, NA, T >::Range (
             const T & obj_ )  [inline]
```

Definition at line 33 of file range.h.

6.657.4 Member Function Documentation

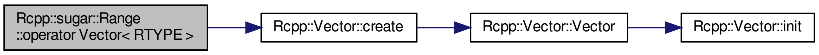
6.657.4.1 operator Vector< RTYPE >()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE > ( )  [inline]
```

Definition at line 35 of file range.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::sugar::Range< RTYPE, NA, T >::current, Rcpp::sugar::Range< RTYPE, NA, T >::max_, Rcpp::sugar::Range< RTYPE, NA, T >::min_, and Rcpp::sugar::Range< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.657.5 Member Data Documentation

6.657.5.1 current

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::current  [private]
```

Definition at line 53 of file range.h.

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), and Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >().

6.657.5.2 max_

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::max_ [private]
```

Definition at line 53 of file range.h.

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), and Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >().

6.657.5.3 min_

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::min_ [private]
```

Definition at line 53 of file range.h.

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), and Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >().

6.657.5.4 obj

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Range< RTYPE, NA, T >::obj [private]
```

Definition at line 52 of file range.h.

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), and Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/range.h

6.658 Rcpp::sugar::Range< RTYPE, false, T > Class Template Reference

```
#include <range.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Range (const T &obj_)
- operator Vector< RTYPE > ()

Private Attributes

- const T & obj
- STORAGE min_
- STORAGE max_
- STORAGE current

6.658.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Range< RTYPE, false, T >

Definition at line 58 of file range.h.

6.658.2 Member Typedef Documentation

6.658.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Range< RTYPE, false, T >::STORAGE
```

Definition at line 60 of file range.h.

6.658.3 Constructor & Destructor Documentation

6.658.3.1 Range()

```
template<int RTYPE, typename T >
Rcpp::sugar::Range< RTYPE, false, T >::Range (
             const T & obj_ )  [inline]
```

Definition at line 62 of file range.h.

6.658.4 Member Function Documentation

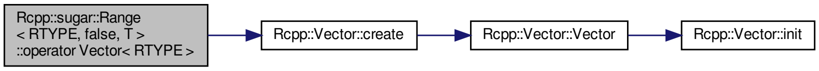
6.658.4.1 operator Vector< RTYPE >()

```
template<int RTYPE, typename T >
Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE > ( )  [inline]
```

Definition at line 64 of file range.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::sugar::Range< RTYPE, NA, T >::current, Rcpp::sugar::Range< RTYPE, NA, T >::max_, Rcpp::sugar::Range< RTYPE, NA, T >::min_, and Rcpp::sugar::Range< RTYPE, NA, T >::obj.

Here is the call graph for this function:



6.658.5 Member Data Documentation

6.658.5.1 current

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::current  [private]
```

Definition at line 79 of file range.h.

6.658.5.2 max_

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::max_  [private]
```

Definition at line 79 of file range.h.

6.658.5.3 min_

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::min_  [private]
```

Definition at line 79 of file range.h.

6.658.5.4 obj

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Range< RTYPE, false, T >::obj  [private]
```

Definition at line 78 of file range.h.

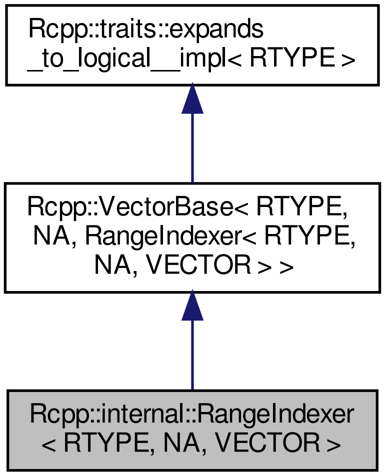
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/range.h

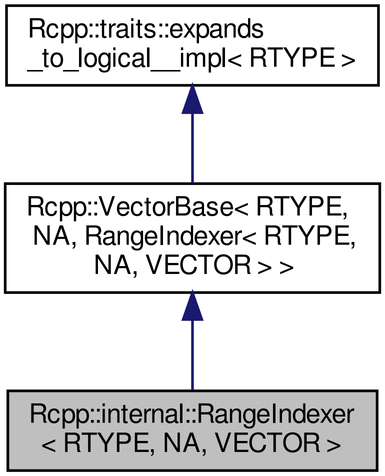
6.659 Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR > Class Template Reference

```
#include <RangeIndexer.h>
```

Inheritance diagram for Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >:



Collaboration diagram for Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, RangeIndexer< RTYPE, NA, VECTOR > >
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_

6.659.1 Detailed Description

template<int RTYPE, bool NA, typename VECTOR>
class Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >

Definition at line 53 of file RangeIndexer.h.

6.659.2 Member Typedef Documentation

6.659.2.1 iterator

```
template<int RTYPE, bool NA, typename VECTOR >
typedef VECTOR::iterator Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::iterator
```

Definition at line 56 of file RangeIndexer.h.

6.659.2.2 Proxy

```
template<int RTYPE, bool NA, typename VECTOR >
typedef VECTOR::Proxy Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::Proxy
```

Definition at line 55 of file RangeIndexer.h.

6.659.3 Constructor & Destructor Documentation

6.659.3.1 RangeIndexer()

```
template<int RTYPE, bool NA, typename VECTOR >
Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::RangeIndexer (
             VECTOR & vec_,
             const Rcpp::Range & range_ )  [inline]
```

Definition at line 58 of file RangeIndexer.h.

6.659.4 Member Function Documentation

6.659.4.1 operator∗=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator*= (
             const Rcpp::VectorBase< RTYPE, NA_, T > & x )  [inline]
```

Definition at line 72 of file RangeIndexer.h.

References UNROLL_LOOP.

6.659.4.2 operator+=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator+= (
             const Rcpp::VectorBase< RTYPE, NA_, T > & x )  [inline]
```

Definition at line 67 of file RangeIndexer.h.

References UNROLL_LOOP.

6.659.4.3 operator-=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator-= (
             const Rcpp::VectorBase< RTYPE, NA_, T > & x )  [inline]
```

Definition at line 77 of file RangeIndexer.h.

References UNROLL_LOOP.

6.659.4.4 operator/=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator/= (
             const Rcpp::VectorBase< RTYPE, NA_, T > & x )  [inline]
```

Definition at line 82 of file RangeIndexer.h.

References UNROLL_LOOP.

6.659.4.5 operator=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator= (
             const Rcpp::VectorBase< RTYPE, NA_, T > & x )  [inline]
```

Definition at line 62 of file RangeIndexer.h.

References UNROLL_LOOP.

6.659.4.6 operator[]()

```
template<int RTYPE, bool NA, typename VECTOR >
Proxy Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 86 of file RangeIndexer.h.

References Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::start.

6.659.4.7 size()

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size ( ) const  [inline]
```

Definition at line 90 of file RangeIndexer.h.

References Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size_.

6.659.5 Member Data Documentation

6.659.5.1 size_

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size_  [private]
```

Definition at line 96 of file RangeIndexer.h.

Referenced by Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size().

6.659.5.2 start

```
template<int RTYPE, bool NA, typename VECTOR >
iterator Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::start  [private]
```

Definition at line 95 of file RangeIndexer.h.

Referenced by Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/RangeIndexer.h

6.660 Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression Struct Reference

```
#include <VectorBase.h>
```

6.660.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression

Definition at line 31 of file VectorBase.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

6.661 Rcpp::ReferenceInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

Public Types

- typedef T & reference

Public Member Functions

- ReferenceInputParameter (SEXP x_)
- operator reference ()

Private Attributes

- T obj

6.661.1 Detailed Description

template<typename T>
class Rcpp::ReferenceInputParameter< T >

Definition at line 43 of file InputParameter.h.

6.661.2 Member Typedef Documentation

6.661.2.1 reference

```
template<typename T >
typedef T& Rcpp::ReferenceInputParameter< T >::reference
```

Definition at line 45 of file InputParameter.h.

6.661.3 Constructor & Destructor Documentation

6.661.3.1 ReferenceInputParameter()

```
template<typename T >
Rcpp::ReferenceInputParameter< T >::ReferenceInputParameter (
              SEXP x_ )  [inline]
```

Definition at line 46 of file InputParameter.h.

6.661.4 Member Function Documentation

6.661.4.1 operator reference()

```
template<typename T >
Rcpp::ReferenceInputParameter< T >::operator reference ( ) [inline]
```

Definition at line 48 of file InputParameter.h.

References Rcpp::ReferenceInputParameter< T >::obj.

6.661.5 Member Data Documentation

6.661.5.1 obj

```
template<typename T >
T Rcpp::ReferenceInputParameter< T >::obj [private]
```

Definition at line 51 of file InputParameter.h.

Referenced by Rcpp::ReferenceInputParameter< T >::operator reference().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

6.662 Rcpp::traits::remove_const< _Tp > Struct Template Reference

const-volatile modifications [4.7.1].

```
#include <remove_const.h>
```

Public Types

- typedef _Tp type

6.662.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_const< _Tp >

const-volatile modifications [4.7.1].

Definition at line 30 of file remove_const.h.

6.662.2 Member Typedef Documentation

6.662.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_const< _Tp >::type
```

Definition at line 31 of file remove_const.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove_const.h

6.663 Rcpp::traits::remove_const< _Tp const > Struct Template Reference

```
#include <remove_const.h>
```

Public Types

- typedef _Tp type

6.663.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_const< _Tp const >

Definition at line 34 of file remove_const.h.

6.663.2 Member Typedef Documentation

6.663.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_const< _Tp const >::type
```

Definition at line 35 of file remove_const.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove_const.h

6.664 Rcpp::traits::remove_const_and_reference< T > Struct Template Reference

```
#include <remove_const_and_reference.h>
```

Public Types

- typedef remove_const< typename remove_reference< T >::type >::type type

6.664.1 Detailed Description

template<typename T>
struct Rcpp::traits::remove_const_and_reference< T >

Definition at line 29 of file remove_const_and_reference.h.

6.664.2 Member Typedef Documentation

6.664.2.1 type

```
template<typename T >
typedef remove_const< typename remove_reference<T>::type >::type Rcpp::traits::remove_const_and_reference<
T >::type
```

Definition at line 30 of file remove_const_and_reference.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove_const_and_reference.h

6.665 Rcpp::traits::remove_reference< _Tp > Struct Template Reference

reference modifications [4.7.2].

```
#include <remove_reference.h>
```

Public Types

- typedef _Tp type

6.665.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_reference< _Tp >

reference modifications [4.7.2].

Definition at line 30 of file remove_reference.h.

6.665.2 Member Typedef Documentation

6.665.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_reference< _Tp >::type
```

Definition at line 31 of file remove_reference.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove_reference.h

6.666 Rcpp::traits::remove_reference< _Tp & > Struct Template Reference

```
#include <remove_reference.h>
```

Public Types

- typedef _Tp type

6.666.1 Detailed Description

template<typename _Tp>
struct Rcpp::traits::remove_reference< _Tp & >

Definition at line 34 of file remove_reference.h.

6.666.2 Member Typedef Documentation

6.666.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_reference< _Tp & >::type
```

Definition at line 35 of file remove_reference.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove_reference.h

6.667 Rcpp::sugar::RemoveFromSet< SET > Class Template Reference

```
#include <setdiff.h>
```

Public Member Functions

- RemoveFromSet (SET &set_)
- template<typename T >
  void operator() (T value)

Private Attributes

- SET & set

6.667.1 Detailed Description

template<typename SET>
class Rcpp::sugar::RemoveFromSet< SET >

Definition at line 29 of file setdiff.h.

6.667.2 Constructor & Destructor Documentation

6.667.2.1 RemoveFromSet()

```
template<typename SET >
Rcpp::sugar::RemoveFromSet< SET >::RemoveFromSet (
            SET & set_ )  [inline]
```

Definition at line 31 of file setdiff.h.

6.667.3 Member Function Documentation

6.667.3.1 operator()()

```
template<typename SET >
template<typename T >
void Rcpp::sugar::RemoveFromSet< SET >::operator() (
             T value )  [inline]
```

Definition at line 34 of file setdiff.h.

References Rcpp::sugar::RemoveFromSet< SET >::set.

6.667.4 Member Data Documentation

6.667.4.1 set

```
template<typename SET >
SET& Rcpp::sugar::RemoveFromSet< SET >::set  [private]
```

Definition at line 39 of file setdiff.h.

Referenced by Rcpp::sugar::RemoveFromSet< SET >::operator()().

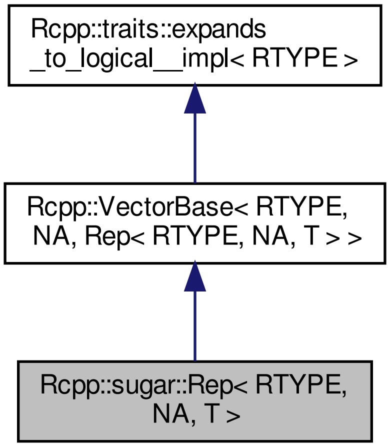
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

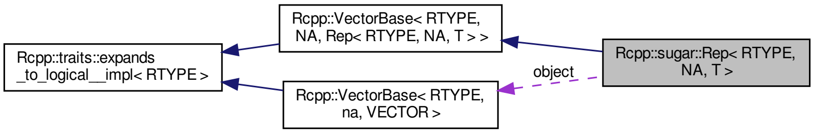
6.668 Rcpp::sugar::Rep< RTYPE, NA, T > Class Template Reference

```
#include <rep.h>
```

Inheritance diagram for Rcpp::sugar::Rep< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > >
Rcpp::sugar::Rep< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Rep< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
object
Rcpp::sugar::Rep< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Rep (const VEC_TYPE &object_, R_xlen_t times_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- R_xlen_t times
- R_xlen_t n

6.668.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep< RTYPE, NA, T >

Definition at line 29 of file rep.h.

6.668.2 Member Typedef Documentation

6.668.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rep< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rep.h.

6.668.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rep.h.

6.668.3 Constructor & Destructor Documentation

6.668.3.1 Rep()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rep< RTYPE, NA, T >::Rep (
             const VEC_TYPE & object_,
             R_xlen_t times_ )  [inline]
```

Definition at line 34 of file rep.h.

6.668.4 Member Function Documentation

6.668.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Rep< RTYPE, NA, T >::operator[ ] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n.

6.668.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 40 of file rep.h.

References Rcpp::sugar::Rep< RTYPE, NA, T >::n, and Rcpp::sugar::Rep< RTYPE, NA, T >::times.

6.668.5 Member Data Documentation

6.668.5.1 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::n  [private]
```

Definition at line 44 of file rep.h.

Referenced by Rcpp::sugar::Rep< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Rep< RTYPE, NA, T >::size().

6.668.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Rep< RTYPE, NA, T >::object  [private]
```

Definition at line 43 of file rep.h.

6.668.5.3 times

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::times  [private]
```

Definition at line 44 of file rep.h.

Referenced by Rcpp::sugar::Rep< RTYPE, NA, T >::size().

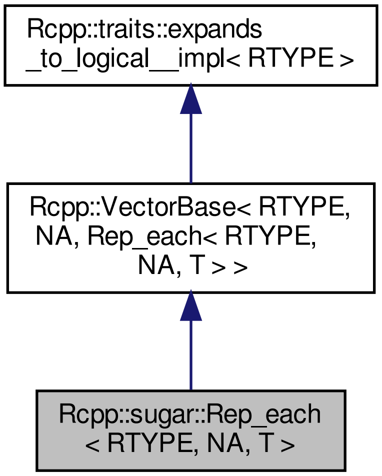
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep.h

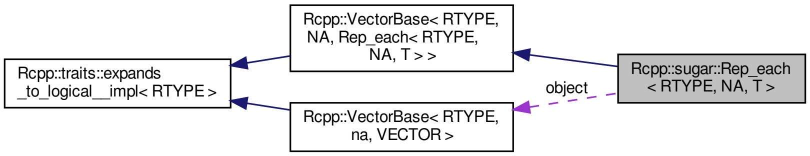
6.669 Rcpp::sugar::Rep_each< RTYPE, NA, T > Class Template Reference

```
#include <rep_each.h>
```

Inheritance diagram for Rcpp::sugar::Rep_each< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Rep_each< RTYPE, NA, T > >
Rcpp::sugar::Rep_each< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Rep_each< RTYPE, NA, T >:


Rcpp::VectorBase< RTYPE, NA, Rep_each< RTYPE, NA, T > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::sugar::Rep_each< RTYPE, NA, T >
object
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Rep_each (const VEC_TYPE &object_, R_xlen_t times_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- R_xlen_t times
- R_xlen_t n

6.669.1 Detailed Description

template< int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep_each< RTYPE, NA, T >

Definition at line 29 of file rep_each.h.

6.669.2 Member Typedef Documentation

6.669.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rep_each< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rep_each.h.

6.669.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep_each< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rep_each.h.

6.669.3 Constructor & Destructor Documentation

6.669.3.1 Rep_each()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rep_each< RTYPE, NA, T >::Rep_each (
             const VEC_TYPE & object_,
             R_xlen_t times_ )  [inline]
```

Definition at line 34 of file rep_each.h.

6.669.4 Member Function Documentation

6.669.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Rep_each< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file rep_each.h.

References Rcpp::sugar::Rep_each< RTYPE, NA, T >::times.

6.669.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 40 of file rep_each.h.

References Rcpp::sugar::Rep_each< RTYPE, NA, T >::n, and Rcpp::sugar::Rep_each< RTYPE, NA, T >::times.

6.669.5 Member Data Documentation

6.669.5.1 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::n  [private]
```

Definition at line 44 of file rep_each.h.

Referenced by Rcpp::sugar::Rep_each< RTYPE, NA, T >::size().

6.669.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Rep_each< RTYPE, NA, T >::object  [private]
```

Definition at line 43 of file rep_each.h.

6.669.5.3 times

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::times  [private]
```

Definition at line 44 of file rep_each.h.

Referenced by Rcpp::sugar::Rep_each< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Rep_each< RTYPE, NA, T >::size().

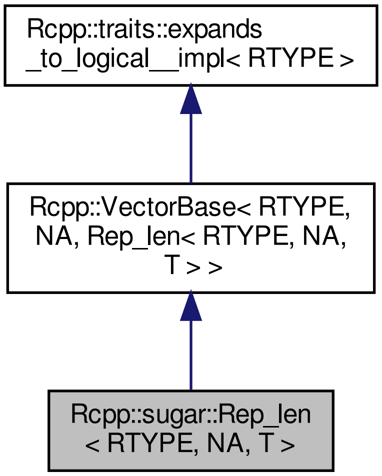
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep_each.h

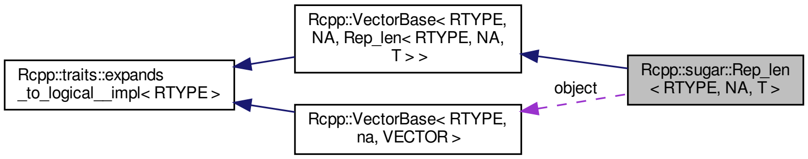
6.670 Rcpp::sugar::Rep_len< RTYPE, NA, T > Class Template Reference

```
#include <rep_len.h>
```

Inheritance diagram for Rcpp::sugar::Rep_len< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Rep_len< RTYPE, NA, T > >
Rcpp::sugar::Rep_len < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Rep_len< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, NA, Rep_len< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
object
Rcpp::sugar::Rep_len < RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Rep_len (const VEC_TYPE &object_, R_xlen_t len_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- R_xlen_t len
- R_xlen_t n

6.670.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep_len< RTYPE, NA, T >

Definition at line 29 of file rep_len.h.

6.670.2 Member Typedef Documentation

6.670.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rep_len< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rep_len.h.

6.670.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep_len< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rep_len.h.

6.670.3 Constructor & Destructor Documentation

6.670.3.1 Rep_len()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rep_len< RTYPE, NA, T >::Rep_len (
            const VEC_TYPE & object_,
            R_xlen_t len_ )  [inline]
```

Definition at line 34 of file rep_len.h.

6.670.4 Member Function Documentation

6.670.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Rep_len< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file rep_len.h.

References Rcpp::sugar::Rep_len< RTYPE, NA, T >::n.

6.670.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 40 of file rep_len.h.

References Rcpp::sugar::Rep_len< RTYPE, NA, T >::len.

6.670.5 Member Data Documentation

6.670.5.1 len

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::len  [private]
```

Definition at line 44 of file rep_len.h.

Referenced by Rcpp::sugar::Rep_len< RTYPE, NA, T >::size().

6.670.5.2 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::n  [private]
```

Definition at line 44 of file rep_len.h.

Referenced by Rcpp::sugar::Rep_len< RTYPE, NA, T >::operator[ ]().

6.670.5.3 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Rep_len< RTYPE, NA, T >::object  [private]
```

Definition at line 43 of file rep_len.h.

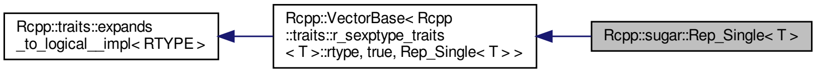
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep_len.h

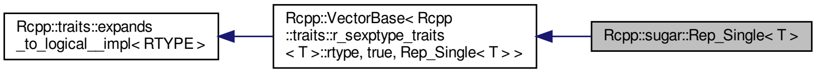
6.671 Rcpp::sugar::Rep_Single< T > Class Template Reference

```
#include <rep.h>
```

Inheritance diagram for Rcpp::sugar::Rep_Single< T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < T >::rtype, true, Rep_Single< T > >
Rcpp::sugar::Rep_Single< T >

Collaboration diagram for Rcpp::sugar::Rep_Single< T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < T >::rtype, true, Rep_Single< T > >
Rcpp::sugar::Rep_Single< T >

Public Member Functions

- Rep_Single (const T &x_, R_xlen_t n_)
- T operator[ ] (R_xlen_t) const
- R_xlen_t size () const

Private Attributes

- const T & x
- R_xlen_t n

Additional Inherited Members

6.671.1 Detailed Description

template<typename T>
class Rcpp::sugar::Rep_Single< T >

Definition at line 48 of file rep.h.

6.671.2 Constructor & Destructor Documentation

6.671.2.1 Rep_Single()

```
template<typename T >
Rcpp::sugar::Rep_Single< T >::Rep_Single (
            const T & x_,
            R_xlen_t n_ )  [inline]
```

Definition at line 54 of file rep.h.

6.671.3 Member Function Documentation

6.671.3.1 operator[]()

```
template<typename T >
T Rcpp::sugar::Rep_Single< T >::operator[] (
             R_xlen_t  ) const  [inline]
```

Definition at line 56 of file rep.h.

References Rcpp::sugar::Rep_Single< T >::x.

6.671.3.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Rep_Single< T >::size ( ) const  [inline]
```

Definition at line 59 of file rep.h.

References Rcpp::sugar::Rep_Single< T >::n.

6.671.4 Member Data Documentation

6.671.4.1 n

```
template<typename T >
R_xlen_t Rcpp::sugar::Rep_Single< T >::n  [private]
```

Definition at line 63 of file rep.h.

Referenced by Rcpp::sugar::Rep_Single< T >::size().

6.671.4.2 x

```
template<typename T >
const T& Rcpp::sugar::Rep_Single< T >::x  [private]
```

Definition at line 62 of file rep.h.

Referenced by Rcpp::sugar::Rep_Single< T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep.h

6.672 Rcpp::result< T > Class Template Reference

```
#include <Module.h>
```

Public Member Functions

- result (T *ptr_)
- operator T* ()

Private Attributes

- T * ptr

6.672.1 Detailed Description

template<typename T>
class Rcpp::result< T >

Definition at line 56 of file Module.h.

6.672.2 Constructor & Destructor Documentation

6.672.2.1 result()

```
template<typename T >
Rcpp::result< T >::result (
            T * ptr_ )  [inline]
```

Definition at line 58 of file Module.h.

6.672.3 Member Function Documentation

6.672.3.1 operator T*()

```
template<typename T >
Rcpp::result< T >::operator T* ( )  [inline]
```

Definition at line 59 of file Module.h.

References Rcpp::result< T >::ptr.

6.672.4 Member Data Documentation

6.672.4.1 ptr

```
template<typename T >
T* Rcpp::result< T >::ptr  [private]
```

Definition at line 61 of file Module.h.

Referenced by Rcpp::result< T >::operator T*().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.673 Rcpp::sugar::median_detail::result< RTYPE > Struct Template Reference

```
#include <median.h>
```

Public Types

- enum { rtype = RTYPE }
- typedef Rcpp::traits::storage_type< RTYPE >::type type

6.673.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::median_detail::result< RTYPE >

Definition at line 35 of file median.h.

6.673.2 Member Typedef Documentation

6.673.2.1 type

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::median_detail::result< RTYPE >↩
::type
```

Definition at line 36 of file median.h.

6.673.3 Member Enumeration Documentation

6.673.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

Enumerator

rtype	

Definition at line 37 of file median.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.674 Rcpp::sugar::median_detail::result< INTSXP > Struct Reference

```
#include <median.h>
```

Public Types

- enum { rtype = REALSXP }
- typedef double type

6.674.1 Detailed Description

Definition at line 41 of file median.h.

6.674.2 Member Typedef Documentation

6.674.2.1 type

```
typedef double Rcpp::sugar::median_detail::result< INTSXP >::type
```

Definition at line 42 of file median.h.

6.674.3 Member Enumeration Documentation

6.674.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 43 of file median.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.675 Rcpp::sugar::median_detail::result< STRSXP > Struct Reference

```
#include <median.h>
```

Public Types

- enum { rtype = STRSXP }
- typedef Rcpp::String type

6.675.1 Detailed Description

Definition at line 47 of file median.h.

6.675.2 Member Typedef Documentation

6.675.2.1 type

```
typedef Rcpp::String Rcpp::sugar::median_detail::result< STRSXP >::type
```

Definition at line 48 of file median.h.

6.675.3 Member Enumeration Documentation

6.675.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 49 of file median.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/median.h

6.676 Rcpp::traits::result_of< T > Struct Template Reference

```
#include <result_of.h>
```

Public Types

- typedef T::result_type type

6.676.1 Detailed Description

template<typename T>
struct Rcpp::traits::result_of< T >

Definition at line 30 of file result_of.h.

6.676.2 Member Typedef Documentation

6.676.2.1 type

```
template<typename T >
typedef T::result_type Rcpp::traits::result_of< T >::type
```

Definition at line 31 of file result_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result_of.h

6.677 Rcpp::traits::result_of< RESULT_TYPE(∗)(INPUT_TYPE) > Struct Template Reference

```
#include <result_of.h>
```

Public Types

- typedef RESULT_TYPE type

6.677.1 Detailed Description

template<typename RESULT_TYPE, typename INPUT_TYPE>
struct Rcpp::traits::result_of< RESULT_TYPE(∗)(INPUT_TYPE) >

Definition at line 35 of file result_of.h.

6.677.2 Member Typedef Documentation

6.677.2.1 type

```
template<typename RESULT_TYPE , typename INPUT_TYPE >
typedef RESULT_TYPE Rcpp::traits::result_of< RESULT_TYPE(∗)(INPUT_TYPE) >::type
```

Definition at line 36 of file result_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result_of.h

6.678 Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2) > Struct Template Reference

```
#include <result_of.h>
```

Public Types

- typedef RESULT_TYPE type

6.678.1 Detailed Description

template<typename RESULT_TYPE, typename U1, typename U2>
struct Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2) >

Definition at line 40 of file result_of.h.

6.678.2 Member Typedef Documentation

6.678.2.1 type

```
template<typename RESULT_TYPE , typename U1 , typename U2 >
typedef RESULT_TYPE Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2) >::type
```

Definition at line 41 of file result_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result_of.h

6.679 Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2, U3) > Struct Template Reference

```
#include <result_of.h>
```

Public Types

- typedef RESULT_TYPE type

6.679.1 Detailed Description

template<typename RESULT_TYPE, typename U1, typename U2, typename U3>
struct Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2, U3) >

Definition at line 45 of file result_of.h.

6.679.2 Member Typedef Documentation

6.679.2.1 type

```
template<typename RESULT_TYPE , typename U1 , typename U2 , typename U3 >
typedef RESULT_TYPE Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2, U3) >::type
```

Definition at line 46 of file result_of.h.

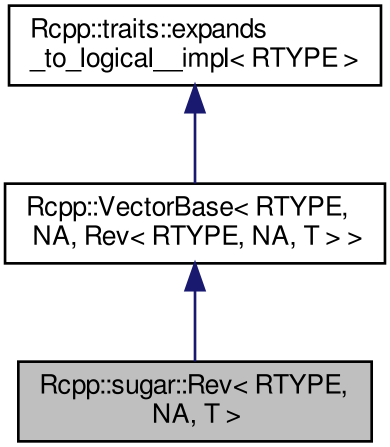
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result_of.h

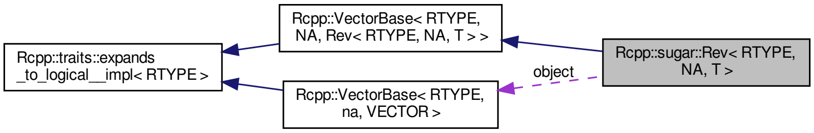
6.680 Rcpp::sugar::Rev< RTYPE, NA, T > Class Template Reference

```
#include <rev.h>
```

Inheritance diagram for Rcpp::sugar::Rev< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Rev< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Rev (const VEC_TYPE &object_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- R_xlen_t n

6.680.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rev< RTYPE, NA, T >

Definition at line 29 of file rev.h.

6.680.2 Member Typedef Documentation

6.680.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rev< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rev.h.

6.680.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rev< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rev.h.

6.680.3 Constructor & Destructor Documentation

6.680.3.1 Rev()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rev< RTYPE, NA, T >::Rev (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 34 of file rev.h.

6.680.4 Member Function Documentation

6.680.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Rev< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file rev.h.

References Rcpp::sugar::Rev< RTYPE, NA, T >::n.

6.680.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rev< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 40 of file rev.h.

References Rcpp::sugar::Rev< RTYPE, NA, T >::n.

6.680.5 Member Data Documentation

6.680.5.1 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rev< RTYPE, NA, T >::n  [private]
```

Definition at line 44 of file rev.h.

Referenced by Rcpp::sugar::Rev< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Rev< RTYPE, NA, T >::size().

6.680.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Rev< RTYPE, NA, T >::object  [private]
```

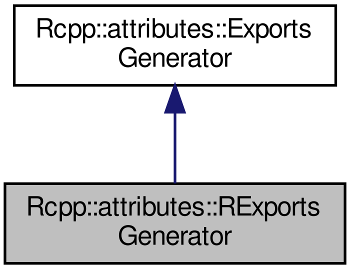
Definition at line 43 of file rev.h.

The documentation for this class was generated from the following file:

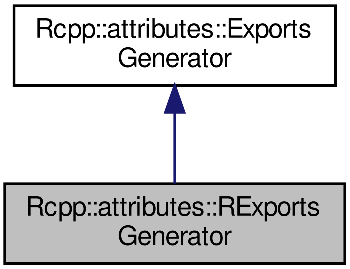
- inst/include/Rcpp/sugar/functions/rev.h

6.681 Rcpp::attributes::RExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::RExportsGenerator:


Rcpp::attributes::Exports Generator
Rcpp::attributes::RExports Generator

Collaboration diagram for Rcpp::attributes::RExportsGenerator:



Public Member Functions

- RExportsGenerator (const std::string &packageDir, const std::string &package, bool registration, const std::string &fileSep)
- virtual void writeBegin ()
- virtual void writeEnd (bool hasPackageInit)
- virtual bool commit (const std::vector< std::string > &includes)

Private Member Functions

- virtual void doWriteFunctions (const SourceFileAttributes &attributes, bool verbose)

Private Attributes

- bool registration_

Additional Inherited Members

6.681.1 Detailed Description

Definition at line 770 of file attributes.cpp.

6.681.2 Constructor & Destructor Documentation

6.681.2.1 RExportsGenerator()

```
Rcpp::attributes::RExportsGenerator::RExportsGenerator (
            const std::string & packageDir,
            const std::string & package,
            bool registration,
            const std::string & fileSep )
```

Definition at line 2449 of file attributes.cpp.

6.681.3 Member Function Documentation

6.681.3.1 commit()

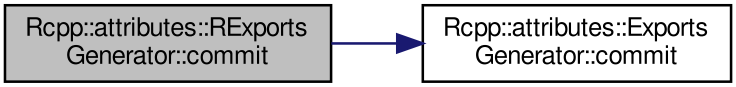
```
bool Rcpp::attributes::RExportsGenerator::commit (
            const std::vector< std::string > & includes )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2549 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::commit().

Here is the call graph for this function:


Rcpp::attributes::RExports Generator::commit
Rcpp::attributes::Exports Generator::commit

6.681.3.2 doWriteFunctions()

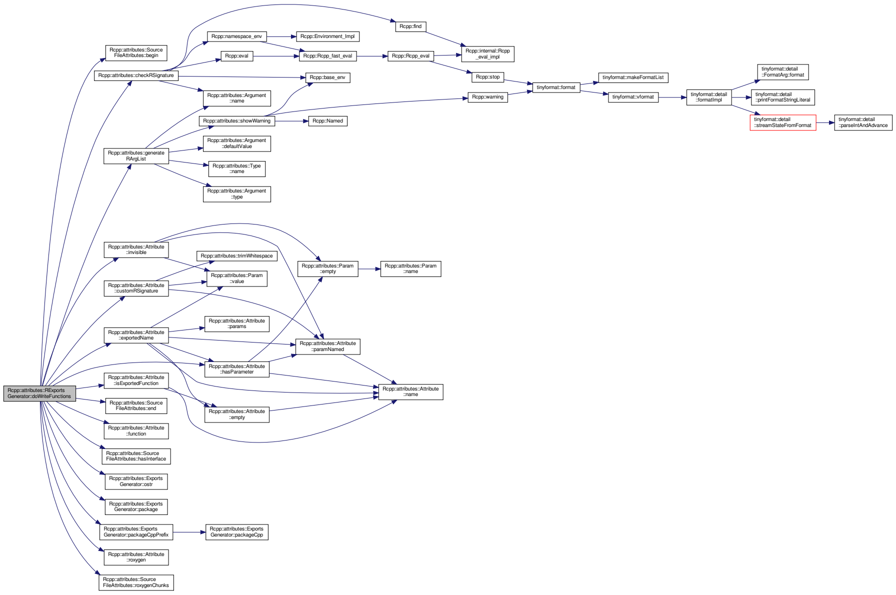
```
void Rcpp::attributes::RExportsGenerator::doWriteFunctions (
            const SourceFileAttributes & attributes,
            bool verbose )  [private], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2461 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::checkRSignature(), Rcpp::attributes::Attribute::customRSignature(), Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::Attribute::exportedName(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::generateRArgList(), Rcpp::attributes::SourceFileAttributes::hasInterface(), Rcpp::attributes::Attribute::hasParameter(), Rcpp::attributes::Attribute::invisible(), Rcpp::attributes::Attribute::isExportedFunction(), Rcpp::attributes::kExportSignature, Rcpp::attributes::kInterfaceR, Rcpp::attributes::ExportsGenerator::ostr(), Rcpp::attributes::ExportsGenerator::package(), Rcpp::attributes::ExportsGenerator::packageCppPrefix(), registration_, Rcpp::attributes::Attribute::roxygen(), and Rcpp::attributes::SourceFileAttributes::roxygenChunks().

Here is the call graph for this function:


Rcpp::find
Rcpp::namespace_env
Rcpp::Environment_Impl
Rcpp::attributes::Source FileAttributes::begin
Rcpp::eval
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::attributes::checkRSignature
Rcpp::base_env
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
Rcpp::warning
Rcpp::attributes::Argument ::name
Rcpp::attributes::showWarning
Rcpp::Named
Rcpp::attributes::Argument ::defaultValue
Rcpp::attributes::generate RArgList
Rcpp::attributes::Type ::name
Rcpp::attributes::Argument ::type
Rcpp::attributes::Attribute ::invisible
Rcpp::attributes::trimWhitespace
Rcpp::attributes::Param ::empty
Rcpp::attributes::Param ::name
Rcpp::attributes::Param ::value
Rcpp::attributes::Attribute ::customRSignature
Rcpp::attributes::Attribute ::params
Rcpp::attributes::Attribute ::exportedName
Rcpp::attributes::Attribute ::paramNamed
Rcpp::attributes::Attribute ::hasParameter
Rcpp::attributes::Attribute ::isExportedFunction
Rcpp::attributes::Attribute ::name
Rcpp::attributes::RExports Generator::doWriteFunctions
Rcpp::attributes::Source FileAttributes::end
Rcpp::attributes::Attribute ::empty
Rcpp::attributes::Attribute ::function
Rcpp::attributes::Source FileAttributes::hasInterface
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::Exports Generator::package
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp
Rcpp::attributes::Attribute ::roxygen
Rcpp::attributes::Source FileAttributes::roxygenChunks

6.681.3.3 writeBegin()

```
virtual void Rcpp::attributes::RExportsGenerator::writeBegin ( )  [inline], [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 777 of file attributes.cpp.

6.681.3.4 writeEnd()

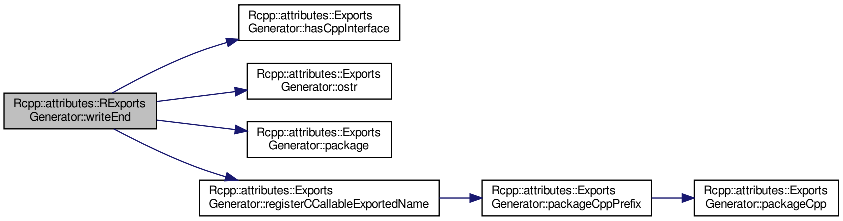
```
void Rcpp::attributes::RExportsGenerator::writeEnd (
            bool hasPackageInit )  [virtual]
```

Implements Rcpp::attributes::ExportsGenerator.

Definition at line 2537 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerator::hasCppInterface(), Rcpp::attributes::ExportsGenerator::ostr(), Rcpp↩ ::attributes::ExportsGenerator::package(), and Rcpp::attributes::ExportsGenerator::registerCCallableExportedName().

Here is the call graph for this function:


Rcpp::attributes::RExports Generator::writeEnd
Rcpp::attributes::Exports Generator::hasCppInterface
Rcpp::attributes::Exports Generator::ostr
Rcpp::attributes::Exports Generator::package
Rcpp::attributes::Exports Generator::registerCCallableExportedName
Rcpp::attributes::Exports Generator::packageCppPrefix
Rcpp::attributes::Exports Generator::packageCpp

6.681.4 Member Data Documentation

6.681.4.1 registration_

```
bool Rcpp::attributes::RExportsGenerator::registration_  [private]
```

Definition at line 785 of file attributes.cpp.

Referenced by doWriteFunctions().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.682 Rcpp::RNGScope Class Reference

```
#include <RNGScope.h>
```

Public Member Functions

- RNGScope ()
- ∼RNGScope ()

6.682.1 Detailed Description

Definition at line 27 of file RNGScope.h.

6.682.2 Constructor & Destructor Documentation

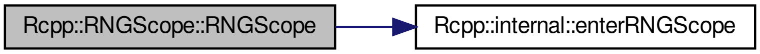
6.682.2.1 RNGScope()

```
Rcpp::RNGScope::RNGScope ( )  [inline]
```

Definition at line 29 of file RNGScope.h.

References Rcpp::internal::enterRNGScope().

Here is the call graph for this function:



6.682.2.2 ∼RNGScope()

```
Rcpp::RNGScope::∼RNGScope ( )  [inline]
```

Definition at line 30 of file RNGScope.h.

References Rcpp::internal::exitRNGScope().

Here is the call graph for this function:

Rcpp::RNGScope::~RNGScope
Rcpp::internal::exitRNGScope

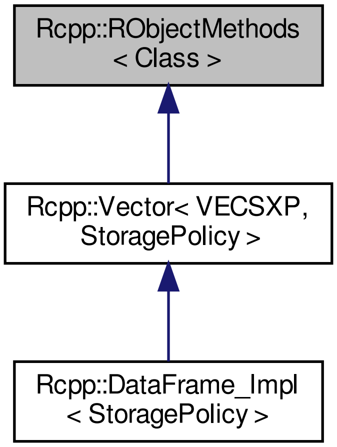
The documentation for this class was generated from the following file:

- inst/include/Rcpp/RNGScope.h

6.683 Rcpp::RObjectMethods< Class > Class Template Reference

```
#include <RObjectMethods.h>
```

Inheritance diagram for Rcpp::RObjectMethods< Class >:


Rcpp::RObjectMethods < Class >
Rcpp::Vector< VECSXP, StoragePolicy >
Rcpp::DataFrame_Impl < StoragePolicy >

Public Member Functions

- bool isNULL () const
- int sexp_type () const
- bool isObject () const
- bool isS4 () const

6.683.1 Detailed Description

template<typename Class>
class Rcpp::RObjectMethods< Class >

Definition at line 24 of file RObjectMethods.h.

6.683.2 Member Function Documentation

6.683.2.1 isNULL()

```
template<typename Class >
bool Rcpp::RObjectMethods< Class >::isNULL ( ) const  [inline]
```

Definition at line 27 of file RObjectMethods.h.

6.683.2.2 isObject()

```
template<typename Class >
bool Rcpp::RObjectMethods< Class >::isObject ( ) const  [inline]
```

Definition at line 35 of file RObjectMethods.h.

6.683.2.3 isS4()

```
template<typename Class >
bool Rcpp::RObjectMethods< Class >::isS4 ( ) const  [inline]
```

Definition at line 39 of file RObjectMethods.h.

6.683.2.4 sexp_type()

```
template<typename Class >
int Rcpp::RObjectMethods< Class >::sexp_type ( ) const  [inline]
```

Definition at line 31 of file RObjectMethods.h.

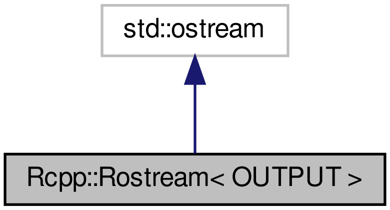
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/RObjectMethods.h

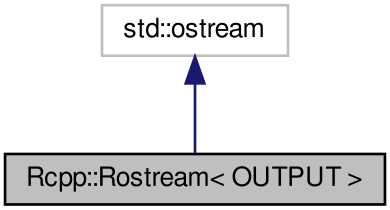
6.684 Rcpp::Rostream< OUTPUT > Class Template Reference

```
#include <Rstreambuf.h>
```

Inheritance diagram for Rcpp::Rostream< OUTPUT >:



Collaboration diagram for Rcpp::Rostream< OUTPUT >:



Public Member Functions

- Rostream ()

Private Types

- typedef Rstreambuf< OUTPUT > Buffer

Private Attributes

- Buffer buf

6.684.1 Detailed Description

template<bool OUTPUT>
class Rcpp::Rostream< OUTPUT >

Definition at line 45 of file Rstreambuf.h.

6.684.2 Member Typedef Documentation

6.684.2.1 Buffer

```
template<bool OUTPUT>
typedef Rstreambuf<OUTPUT> Rcpp::Rostream< OUTPUT >::Buffer  [private]
```

Definition at line 46 of file Rstreambuf.h.

6.684.3 Constructor & Destructor Documentation

6.684.3.1 Rostream()

```
template<bool OUTPUT>
Rcpp::Rostream< OUTPUT >::Rostream ( )  [inline]
```

Definition at line 49 of file Rstreambuf.h.

6.684.4 Member Data Documentation

6.684.4.1 buf

```
template<bool OUTPUT>
Buffer Rcpp::Rostream< OUTPUT >::buf  [private]
```

Definition at line 47 of file Rstreambuf.h.

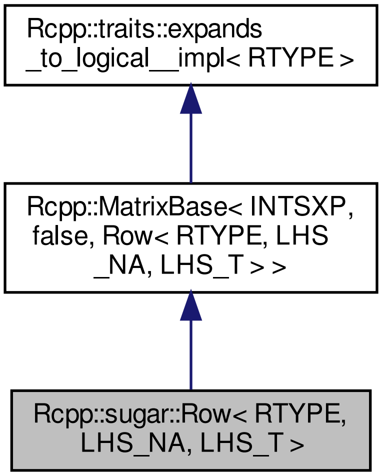
The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/Rstreambuf.h

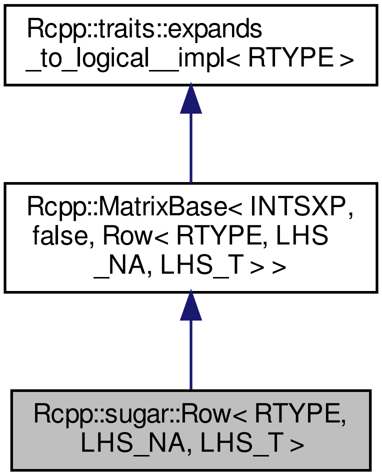
6.685 Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T > Class Template Reference

```
#include <row.h>
```

Inheritance diagram for Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >:



Collaboration diagram for Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >:



Public Types

- typedef Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE

Public Member Functions

- Row (const LHS_TYPE &lhs)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Attributes

- int nr
- int nc

Additional Inherited Members

6.685.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >

Definition at line 29 of file row.h.

6.685.2 Member Typedef Documentation

6.685.2.1 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 35 of file row.h.

6.685.3 Constructor & Destructor Documentation

6.685.3.1 Row()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::Row (
              const LHS_TYPE & lhs )  [inline]
```

Definition at line 37 of file row.h.

6.685.4 Member Function Documentation

6.685.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::ncol ( ) const  [inline]
```

Definition at line 45 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nc.

6.685.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nrow ( ) const  [inline]
```

Definition at line 44 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nr.

6.685.4.3 operator()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 39 of file row.h.

6.685.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::size ( ) const  [inline]
```

Definition at line 43 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nc, and Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nr.

6.685.5 Member Data Documentation

6.685.5.1 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nc  [private]
```

Definition at line 48 of file row.h.

Referenced by Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::ncol(), and Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::size().

6.685.5.2 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nr  [private]
```

Definition at line 48 of file row.h.

Referenced by Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nrow(), and Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::size().

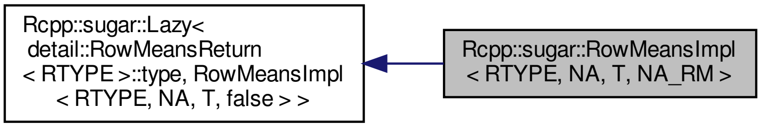
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/row.h

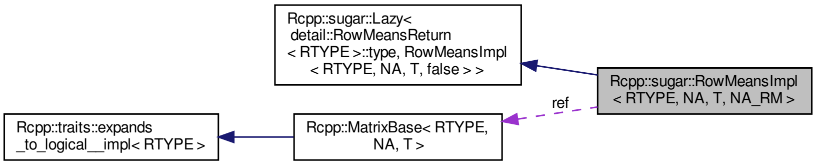
6.686 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::RowMeansReturn < RTYPE >::type, RowMeansImpl < RTYPE, NA, T, false > >
Rcpp::sugar::RowMeansImpl < RTYPE, NA, T, NA_RM >

Collaboration diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::RowMeansReturn < RTYPE >::type, RowMeansImpl < RTYPE, NA, T, false > >
Rcpp::sugar::RowMeansImpl < RTYPE, NA, T, NA_RM >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- RowMeansImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::RowMeansReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.686.1 Detailed Description

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >

Definition at line 507 of file rowSums.h.

6.686.2 Member Typedef Documentation

6.686.2.1 return_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::RowMeansReturn<RTYPE> Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::return_traits
[private]
```

Definition at line 512 of file rowSums.h.

6.686.2.2 return_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return_traits::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::return_vector  [private]
```

Definition at line 513 of file rowSums.h.

6.686.2.3 stored_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T,
NA_RM >::stored_type  [private]
```

Definition at line 514 of file rowSums.h.

6.686.3 Constructor & Destructor Documentation

6.686.3.1 RowMeansImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::RowMeansImpl (
             const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 517 of file rowSums.h.

6.686.4 Member Function Documentation

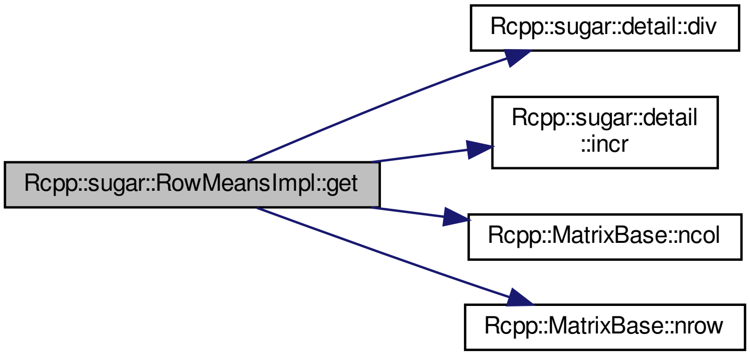
6.686.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get ( ) const  [inline]
```

Definition at line 521 of file rowSums.h.

References Rcpp::sugar::detail::div(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.686.5 Member Data Documentation

6.686.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::ref  [private]
```

Definition at line 510 of file rowSums.h.

Referenced by Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get().

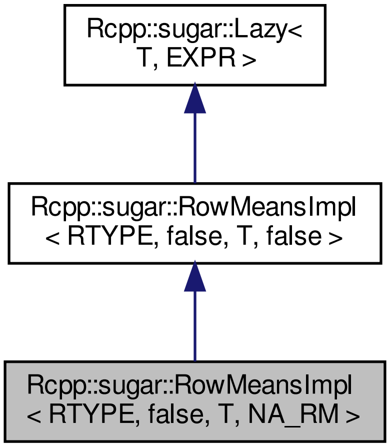
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

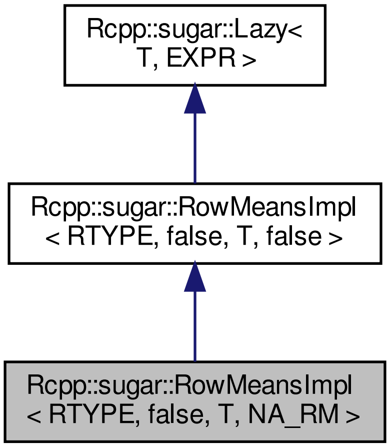
6.687 Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, NA_RM >

Collaboration diagram for Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, false >
Rcpp::sugar::RowMeansImpl < RTYPE, false, T, NA_RM >

Additional Inherited Members

6.687.1 Detailed Description

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >

Definition at line 706 of file rowSums.h.

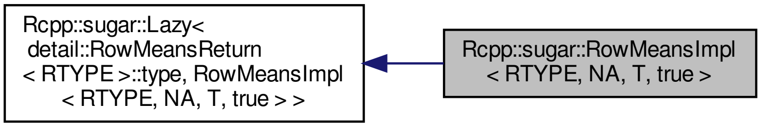
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

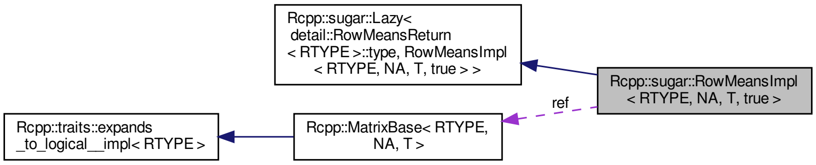
6.688 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::RowMeansReturn < RTYPE >::type, RowMeansImpl < RTYPE, NA, T, true > >
Rcpp::sugar::RowMeansImpl < RTYPE, NA, T, true >

Collaboration diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::RowMeansReturn < RTYPE >::type, RowMeansImpl < RTYPE, NA, T, true > >
Rcpp::sugar::RowMeansImpl < RTYPE, NA, T, true >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- RowMeansImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::RowMeansReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.688.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >

Definition at line 603 of file rowSums.h.

6.688.2 Member Typedef Documentation

6.688.2.1 return_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::RowMeansReturn<RTYPE> Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 608 of file rowSums.h.

6.688.2.2 return_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::return_vector  [private]
```

Definition at line 609 of file rowSums.h.

6.688.2.3 stored_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T,
true >::stored_type  [private]
```

Definition at line 610 of file rowSums.h.

6.688.3 Constructor & Destructor Documentation

6.688.3.1 RowMeansImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::RowMeansImpl (
             const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 613 of file rowSums.h.

6.688.4 Member Function Documentation

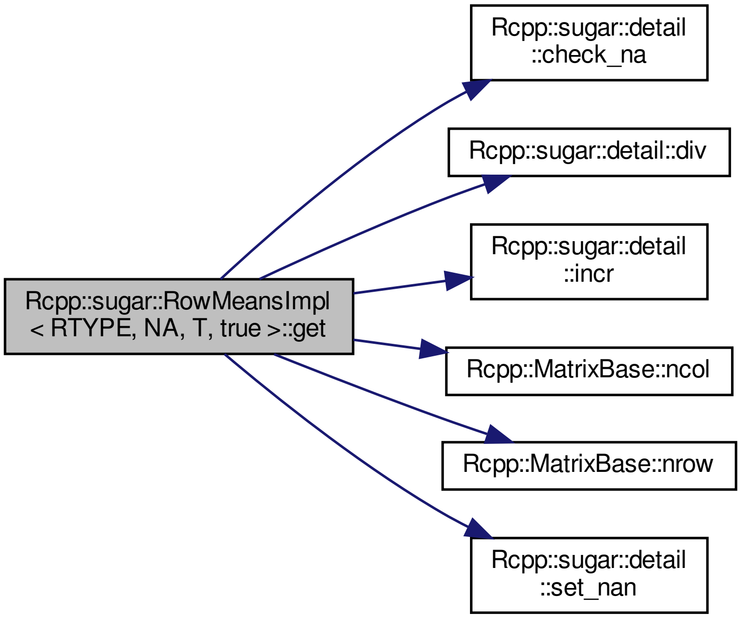
6.688.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get ( ) const  [inline]
```

Definition at line 617 of file rowSums.h.

References Rcpp::sugar::detail::check_na(), Rcpp::sugar::detail::div(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::ref, and Rcpp::sugar::detail::set_nan().

Here is the call graph for this function:



6.688.5 Member Data Documentation

6.688.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::ref  [private]
```

Definition at line 606 of file rowSums.h.

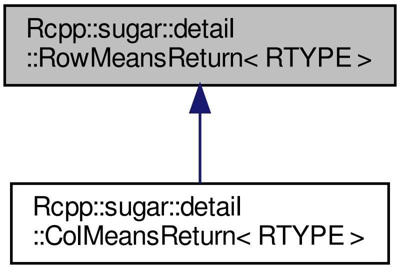
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.689 Rcpp::sugar::detail::RowMeansReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::RowMeansReturn< RTYPE >:



Public Types

- enum { rtype = REALSXP }
- typedef Vector< REALSXP > type

6.689.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::RowMeansReturn< RTYPE >

Definition at line 103 of file rowSums.h.

6.689.2 Member Typedef Documentation

6.689.2.1 type

```
template<int RTYPE>
typedef Vector<REALSXP> Rcpp::sugar::detail::RowMeansReturn< RTYPE >::type
```

Definition at line 104 of file rowSums.h.

6.689.3 Member Enumeration Documentation

6.689.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

Enumerator

rtype	

Definition at line 105 of file rowSums.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.690 Rcpp::sugar::detail::RowMeansReturn< CPLXSXP > Struct Reference

```
#include <rowSums.h>
```

Public Types

- enum { rtype = CPLXSXP }
- typedef Vector< CPLXSXP > type

6.690.1 Detailed Description

Definition at line 109 of file rowSums.h.

6.690.2 Member Typedef Documentation

6.690.2.1 type

```
typedef Vector<CPLXSXP> Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >::type
```

Definition at line 110 of file rowSums.h.

6.690.3 Member Enumeration Documentation

6.690.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 111 of file rowSums.h.

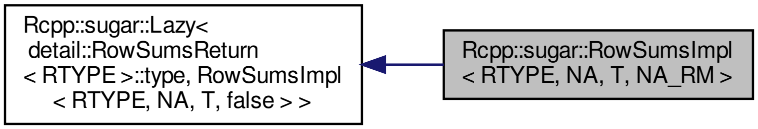
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

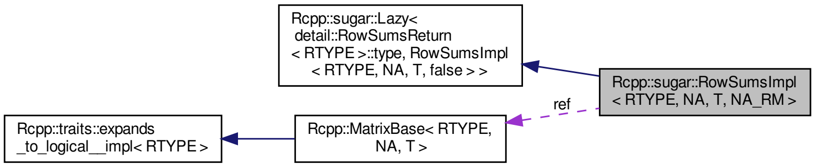
6.691 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::RowSumsReturn < RTYPE >::type, RowSumsImpl < RTYPE, NA, T, false > >
Rcpp::sugar::RowSumsImpl < RTYPE, NA, T, NA_RM >

Collaboration diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >:


Rcpp::sugar::Lazy< detail::RowSumsReturn < RTYPE >::type, RowSumsImpl < RTYPE, NA, T, false > >
Rcpp::sugar::RowSumsImpl < RTYPE, NA, T, NA_RM >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- RowSumsImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::RowSumsReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.691.1 Detailed Description

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >

Definition at line 128 of file rowSums.h.

6.691.2 Member Typedef Documentation

6.691.2.1 return_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::RowSumsReturn<RTYPE> Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::return_traits
[private]
```

Definition at line 133 of file rowSums.h.

6.691.2.2 return_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return_traits::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::return_vector  [private]
```

Definition at line 134 of file rowSums.h.

6.691.2.3 stored_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T,
NA_RM >::stored_type  [private]
```

Definition at line 135 of file rowSums.h.

6.691.3 Constructor & Destructor Documentation

6.691.3.1 RowSumsImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::RowSumsImpl (
             const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 138 of file rowSums.h.

6.691.4 Member Function Documentation

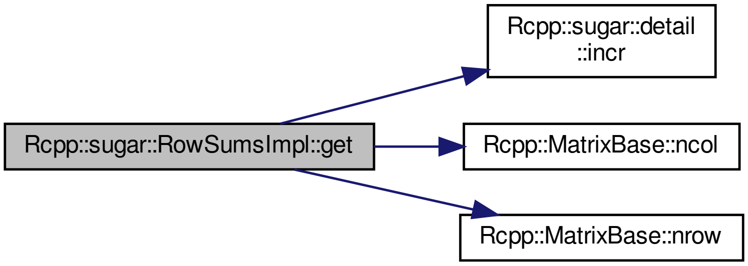
6.691.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get ( ) const  [inline]
```

Definition at line 142 of file rowSums.h.

References Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.691.5 Member Data Documentation

6.691.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::ref  [private]
```

Definition at line 131 of file rowSums.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get(), and Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get().

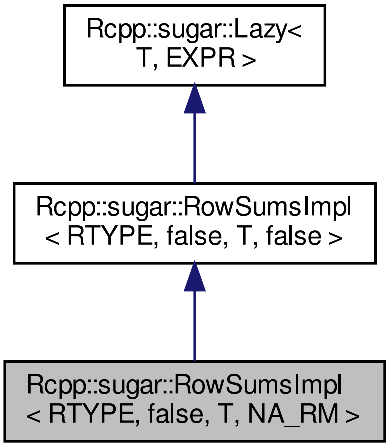
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

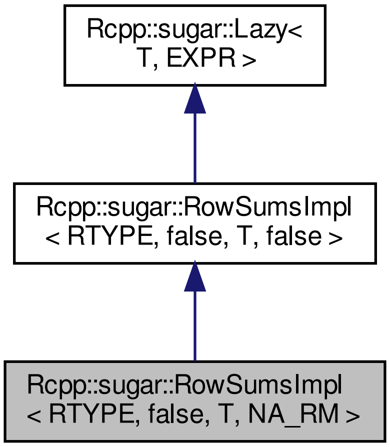
6.692 Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, NA_RM >

Collaboration diagram for Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >:


Rcpp::sugar::Lazy< T, EXPR >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, false >
Rcpp::sugar::RowSumsImpl < RTYPE, false, T, NA_RM >

Additional Inherited Members

6.692.1 Detailed Description

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >

Definition at line 310 of file rowSums.h.

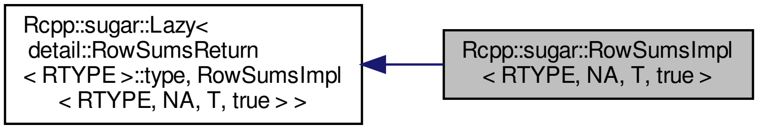
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

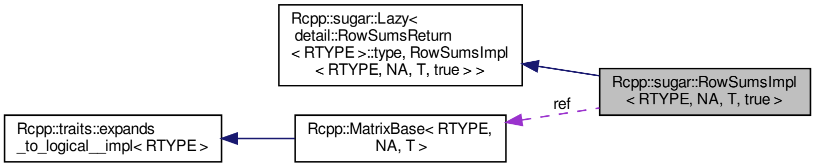
6.693 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::RowSumsReturn < RTYPE >::type, RowSumsImpl < RTYPE, NA, T, true > >
Rcpp::sugar::RowSumsImpl < RTYPE, NA, T, true >

Collaboration diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >:


Rcpp::sugar::Lazy< detail::RowSumsReturn < RTYPE >::type, RowSumsImpl < RTYPE, NA, T, true > >
Rcpp::sugar::RowSumsImpl < RTYPE, NA, T, true >
ref
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::MatrixBase< RTYPE, NA, T >

Public Member Functions

- RowSumsImpl (const MatrixBase< RTYPE, NA, T > &ref_)
- return_vector get () const

Private Types

- typedef detail::RowSumsReturn< RTYPE > return_traits
- typedef return_traits::type return_vector
- typedef traits::storage_type< return_traits::rtype >::type stored_type

Private Attributes

- const MatrixBase< RTYPE, NA, T > & ref

6.693.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >

Definition at line 226 of file rowSums.h.

6.693.2 Member Typedef Documentation

6.693.2.1 return_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::RowSumsReturn<RTYPE> Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 231 of file rowSums.h.

6.693.2.2 return_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::return_vector  [private]
```

Definition at line 232 of file rowSums.h.

6.693.2.3 stored_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T,
true >::stored_type  [private]
```

Definition at line 233 of file rowSums.h.

6.693.3 Constructor & Destructor Documentation

6.693.3.1 RowSumsImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::RowSumsImpl (
            const MatrixBase< RTYPE, NA, T > & ref_ )  [inline]
```

Definition at line 236 of file rowSums.h.

6.693.4 Member Function Documentation

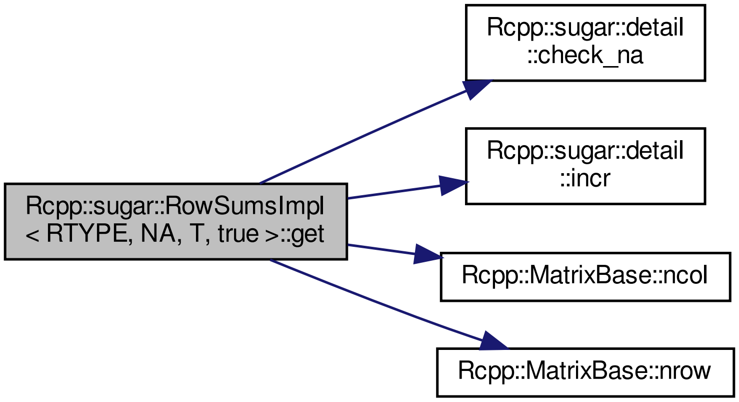
6.693.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get ( ) const  [inline]
```

Definition at line 240 of file rowSums.h.

References Rcpp::sugar::detail::check_na(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >↩::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::ref.

Here is the call graph for this function:



6.693.5 Member Data Documentation

6.693.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::ref  [private]
```

Definition at line 229 of file rowSums.h.

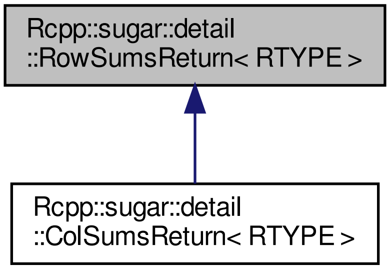
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.694 Rcpp::sugar::detail::RowSumsReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::RowSumsReturn< RTYPE >:



Public Types

- enum { rtype = RTYPE }
- typedef Vector< RTYPE > type

6.694.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::detail::RowSumsReturn< RTYPE >

Definition at line 86 of file rowSums.h.

6.694.2 Member Typedef Documentation

6.694.2.1 type

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::detail::RowSumsReturn< RTYPE >::type
```

Definition at line 87 of file rowSums.h.

6.694.3 Member Enumeration Documentation

6.694.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

Enumerator

rtype	

Definition at line 88 of file rowSums.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

6.695 Rcpp::sugar::detail::RowSumsReturn< LGLSXP > Struct Reference

```
#include <rowSums.h>
```

Public Types

- enum { rtype = INTSXP }
- typedef Vector< INTSXP > type

6.695.1 Detailed Description

Definition at line 92 of file rowSums.h.

6.695.2 Member Typedef Documentation

6.695.2.1 type

```
typedef Vector<INTSXP> Rcpp::sugar::detail::RowSumsReturn< LGLSXP >::type
```

Definition at line 93 of file rowSums.h.

6.695.3 Member Enumeration Documentation

6.695.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	

Definition at line 94 of file rowSums.h.

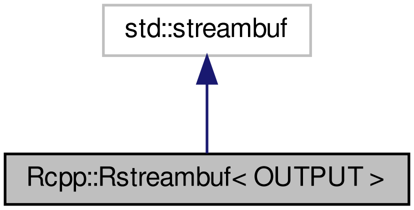
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

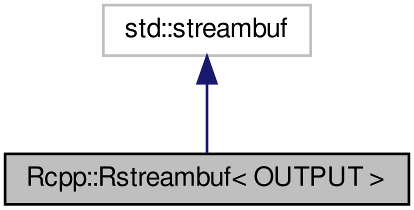
6.696 Rcpp::Rstreambuf< OUTPUT > Class Template Reference

```
#include <Rstreambuf.h>
```

Inheritance diagram for Rcpp::Rstreambuf< OUTPUT >:



Collaboration diagram for Rcpp::Rstreambuf< OUTPUT >:



Public Member Functions

- Rstreambuf ()

Protected Member Functions

- virtual std::streamsize xsputn (const char ∗s, std::streamsize n)
- virtual int overflow (int c=traits_type::eof())
- virtual int sync ()
- std::streamsize xsputn (const char ∗s, std::streamsize num)
- std::streamsize xsputn (const char ∗s, std::streamsize num)
- int overflow (int c)
- int overflow (int c)
- int sync ()
- int sync ()

6.696.1 Detailed Description

template<bool OUTPUT>
class Rcpp::Rstreambuf< OUTPUT >

Definition at line 32 of file Rstreambuf.h.

6.696.2 Constructor & Destructor Documentation

6.696.2.1 Rstreambuf()

```
template<bool OUTPUT>
Rcpp::Rstreambuf< OUTPUT >::Rstreambuf ( )  [inline]
```

Definition at line 34 of file Rstreambuf.h.

6.696.3 Member Function Documentation

6.696.3.1 overflow() [1/3]

```
int Rcpp::Rstreambuf< true >::overflow (
             int c )  [inline], [protected]
```

Definition at line 61 of file Rstreambuf.h.

6.696.3.2 overflow() [2/3]

```
int Rcpp::Rstreambuf< false >::overflow (
             int c )  [inline], [protected]
```

Definition at line 68 of file Rstreambuf.h.

6.696.3.3 overflow() [3/3]

```
template<bool OUTPUT>
virtual int Rcpp::Rstreambuf< OUTPUT >::overflow (
             int c = traits_type::eof() )  [protected], [virtual]
```

6.696.3.4 sync() [1/3]

```
template<bool OUTPUT>
virtual int Rcpp::Rstreambuf< OUTPUT >::sync ( )  [protected], [virtual]
```

6.696.3.5 sync() [2/3]

```
int Rcpp::Rstreambuf< true >::sync ( )  [inline], [protected]
```

Definition at line 76 of file Rstreambuf.h.

6.696.3.6 sync() [3/3]

```
int Rcpp::Rstreambuf< false >::sync ( )  [inline], [protected]
```

Definition at line 80 of file Rstreambuf.h.

6.696.3.7 xsputn() [1/3]

```
template<bool OUTPUT>
virtual std::streamsize Rcpp::Rstreambuf< OUTPUT >::xsputn (
             const char * s,
             std::streamsize n )  [protected], [virtual]
```

6.696.3.8 xsputn() [2/3]

```
std::streamsize Rcpp::Rstreambuf< true >::xsputn (
            const char * s,
            std::streamsize num )  [inline], [protected]
```

Definition at line 52 of file Rstreambuf.h.

6.696.3.9 xsputn() [3/3]

```
std::streamsize Rcpp::Rstreambuf< false >::xsputn (
            const char * s,
            std::streamsize num )  [inline], [protected]
```

Definition at line 56 of file Rstreambuf.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/Rstreambuf.h

6.697 Rcpp::algorithm::helpers::rtype< T > Struct Template Reference

```
#include <algorithm.h>
```

Public Types

- typedef rtype_helper< typename ctype< T >::type >::type type
- typedef rtype_helper< typename ctype< T >::type > helper_type

Static Public Member Functions

- static T NA ()
- static RCPP_CONSTEXPR_FUNC T ZERO ()
- static RCPP_CONSTEXPR_FUNC T ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR_VAR int RTYPE = helper_type::RTYPE

6.697.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::rtype< T >

Definition at line 186 of file algorithm.h.

6.697.2 Member Typedef Documentation

6.697.2.1 helper_type

```
template<typename T >
typedef rtype_helper< typename ctype< T >::type > Rcpp::algorithm::helpers::rtype< T >::helper_type
```

Definition at line 188 of file algorithm.h.

6.697.2.2 type

```
template<typename T >
typedef rtype_helper< typename ctype< T >::type >::type Rcpp::algorithm::helpers::rtype< T >↩
::type
```

Definition at line 187 of file algorithm.h.

6.697.3 Member Function Documentation

6.697.3.1 NA()

```
template<typename T >
static T Rcpp::algorithm::helpers::rtype< T >::NA ( )  [inline], [static]
```

Definition at line 190 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::algorithm::mean(), Rcpp::algorithm::helpers::log::operator()(), Rcpp::algorithm::helpers::exp↩
::operator()(), and Rcpp::algorithm::helpers::sqrt::operator()().

6.697.3.2 ONE()

```
template<typename T >
static RCPP_CONSTEXPR_FUNC T Rcpp::algorithm::helpers::rtype< T >::ONE ( )  [inline], [static]
```

Definition at line 192 of file algorithm.h.

6.697.3.3 ZERO()

```
template<typename T >
static RCPP_CONSTEXPR_FUNC T Rcpp::algorithm::helpers::rtype< T >::ZERO ( )  [inline], [static]
```

Definition at line 191 of file algorithm.h.

6.697.4 Member Data Documentation

6.697.4.1 RTYPE

```
template<typename T >
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype< T >::RTYPE = helper_type::RTYPE  [static]
```

Definition at line 189 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.698 Rcpp::algorithm::helpers::rtype_helper< T > Struct Template Reference

```
#include <algorithm.h>
```

6.698.1 Detailed Description

template<typename T>
struct Rcpp::algorithm::helpers::rtype_helper< T >

Definition at line 164 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.699 Rcpp::algorithm::helpers::rtype_helper< double > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef double type

Static Public Member Functions

- static double NA ()
- static RCPP_CONSTEXPR_FUNC double ZERO ()
- static RCPP_CONSTEXPR_FUNC double ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR_VAR int RTYPE = REALSXP

6.699.1 Detailed Description

Definition at line 168 of file algorithm.h.

6.699.2 Member Typedef Documentation

6.699.2.1 type

```
typedef double Rcpp::algorithm::helpers::rtype_helper< double >::type
```

Definition at line 169 of file algorithm.h.

6.699.3 Member Function Documentation

6.699.3.1 NA()

```
static double Rcpp::algorithm::helpers::rtype_helper< double >::NA ( )  [inline], [static]
```

Definition at line 171 of file algorithm.h.

6.699.3.2 ONE()

```
static RCPP_CONSTEXPR_FUNC double Rcpp::algorithm::helpers::rtype_helper< double >::ONE ( ) [inline],
[static]
```

Definition at line 173 of file algorithm.h.

6.699.3.3 ZERO()

```
static RCPP_CONSTEXPR_FUNC double Rcpp::algorithm::helpers::rtype_helper< double >::ZERO ( )
[inline], [static]
```

Definition at line 172 of file algorithm.h.

6.699.4 Member Data Documentation

6.699.4.1 RTYPE

```
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype_helper< double >::RTYPE = REALSXP [static]
```

Definition at line 170 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.700 Rcpp::algorithm::helpers::rtype_helper< int > Struct Reference

```
#include <algorithm.h>
```

Public Types

- typedef int type

Static Public Member Functions

- static int NA ()
- static RCPP_CONSTEXPR_FUNC int ZERO ()
- static RCPP_CONSTEXPR_FUNC int ONE ()

Static Public Attributes

- static RCPP_CONSTEXPR_VAR int RTYPE = INTSXP

6.700.1 Detailed Description

Definition at line 177 of file algorithm.h.

6.700.2 Member Typedef Documentation

6.700.2.1 type

```
typedef int Rcpp::algorithm::helpers::rtype_helper< int >::type
```

Definition at line 178 of file algorithm.h.

6.700.3 Member Function Documentation

6.700.3.1 NA()

```
static int Rcpp::algorithm::helpers::rtype_helper< int >::NA ( )  [inline], [static]
```

Definition at line 180 of file algorithm.h.

6.700.3.2 ONE()

```
static RCPP_CONSTEXPR_FUNC int Rcpp::algorithm::helpers::rtype_helper< int >::ONE ( )  [inline],
[static]
```

Definition at line 182 of file algorithm.h.

6.700.3.3 ZERO()

```
static RCPP_CONSTEXPR_FUNC int Rcpp::algorithm::helpers::rtype_helper< int >::ZERO ( )  [inline],
[static]
```

Definition at line 181 of file algorithm.h.

6.700.4 Member Data Documentation

6.700.4.1 RTYPE

```
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype_helper< int >::RTYPE = INTSXP  [static]
```

Definition at line 179 of file algorithm.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.701 Rcpp::rule Struct Reference

Public Attributes

- int r_type
- int r_day
- int r_week
- int r_mon
- int_fast32_t r_time

6.701.1 Detailed Description

Definition at line 408 of file date.cpp.

6.701.2 Member Data Documentation

6.701.2.1 r_day

```
int Rcpp::rule::r_day
```

Definition at line 410 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.701.2.2 r_mon

```
int Rcpp::rule::r_mon
```

Definition at line 412 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.701.2.3 r_time

```
int_fast32_t Rcpp::rule::r_time
```

Definition at line 413 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.701.2.4 r_type

```
int Rcpp::rule::r_type
```

Definition at line 409 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

6.701.2.5 r_week

```
int Rcpp::rule::r_week
```

Definition at line 411 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

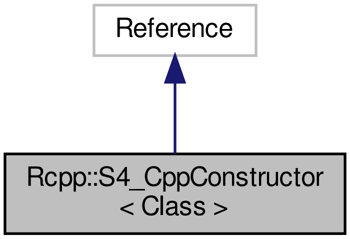
The documentation for this struct was generated from the following file:

- src/date.cpp

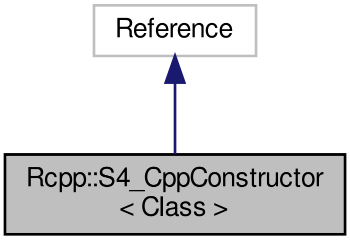
6.702 Rcpp::S4_CppConstructor< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::S4_CppConstructor< Class >:



Collaboration diagram for Rcpp::S4_CppConstructor< Class >:



Public Types

- typedef XPtr< class_Base > XP_Class
- typedef Reference::Storage Storage

Public Member Functions

- S4_CppConstructor (SignedConstructor< Class > ∗m, const XP_Class &class_xp, const std::string &class_name, std::string &buffer)

Private Types

- typedef Reference Base

6.702.1 Detailed Description

template<typename Class>
class Rcpp::S4_CppConstructor< Class >

Definition at line 203 of file Module.h.

6.702.2 Member Typedef Documentation

6.702.2.1 Base

```
template<typename Class >
typedef Reference Rcpp::S4_CppConstructor< Class >::Base  [private]
```

Definition at line 204 of file Module.h.

6.702.2.2 Storage

```
template<typename Class >
typedef Reference::Storage Rcpp::S4_CppConstructor< Class >::Storage
```

Definition at line 207 of file Module.h.

6.702.2.3 XP_Class

```
template<typename Class >
typedef XPtr<class_Base> Rcpp::S4_CppConstructor< Class >::XP_Class
```

Definition at line 206 of file Module.h.

6.702.3 Constructor & Destructor Documentation

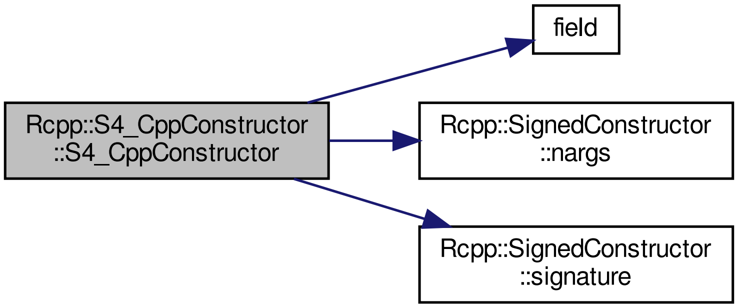
6.702.3.1 S4_CppConstructor()

```
template<typename Class >
Rcpp::S4_CppConstructor< Class >::S4_CppConstructor (
             SignedConstructor< Class > * m,
             const XP_Class & class_xp,
             const std::string & class_name,
             std::string & buffer )  [inline]
```

Definition at line 209 of file Module.h.

References Rcpp::SignedConstructor< Class >::docstring, field(), Rcpp::SignedConstructor< Class >::nargs(), RCPP_DEBUG, and Rcpp::SignedConstructor< Class >::signature().

Here is the call graph for this function:


field
Rcpp::S4_CppConstructor ::S4_CppConstructor
Rcpp::SignedConstructor ::nargs
Rcpp::SignedConstructor ::signature

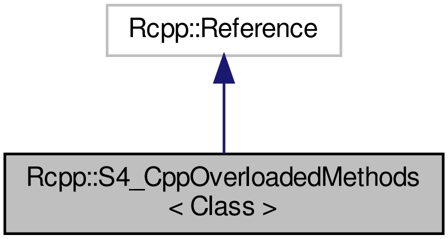
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

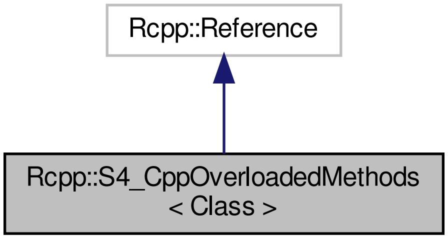
6.703 Rcpp::S4_CppOverloadedMethods< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::S4_CppOverloadedMethods< Class >:



Collaboration diagram for Rcpp::S4_CppOverloadedMethods< Class >:



Public Types

- typedef Rcpp::XPtr< class_Base > XP_Class
- typedef SignedMethod< Class > signed_method_class
- typedef std::vector< signed_method_class ∗ > vec_signed_method

Public Member Functions

- S4_CppOverloadedMethods (vec_signed_method ∗m, const XP_Class &class_xp, const char ∗name, std::string &buffer)

Private Types

- typedef Rcpp::Reference Base

6.703.1 Detailed Description

template<typename Class>
class Rcpp::S4_CppOverloadedMethods< Class >

Definition at line 224 of file Module.h.

6.703.2 Member Typedef Documentation

6.703.2.1 Base

```
template<typename Class >
typedef Rcpp::Reference Rcpp::S4_CppOverloadedMethods< Class >::Base  [private]
```

Definition at line 225 of file Module.h.

6.703.2.2 signed_method_class

```
template<typename Class >
typedef SignedMethod<Class> Rcpp::S4_CppOverloadedMethods< Class >::signed_method_class
```

Definition at line 228 of file Module.h.

6.703.2.3 vec_signed_method

```
template<typename Class >
typedef std::vector<signed_method_class*> Rcpp::S4_CppOverloadedMethods< Class >::vec_signed_method
```

Definition at line 229 of file Module.h.

6.703.2.4 XP_Class

```
template<typename Class >
typedef Rcpp::XPtr<class_Base> Rcpp::S4_CppOverloadedMethods< Class >::XP_Class
```

Definition at line 227 of file Module.h.

6.703.3 Constructor & Destructor Documentation

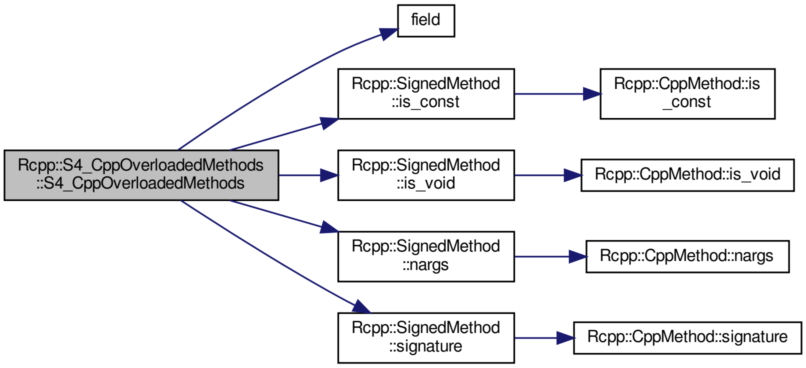
6.703.3.1 S4_CppOverloadedMethods()

```
template<typename Class >
Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods (
             vec_signed_method * m,
             const XP_Class & class_xp,
             const char * name,
             std::string & buffer )  [inline]
```

Definition at line 231 of file Module.h.

References Rcpp::SignedMethod< Class >::docstring, field(), Rcpp::SignedMethod< Class >::is_const(), Rcpp::SignedMethod< Class >::is_void(), Rcpp::SignedMethod< Class >::nargs(), and Rcpp::SignedMethod< Class >::signature().

Here is the call graph for this function:


field
Rcpp::SignedMethod ::is_const
Rcpp::CppMethod::is _const
Rcpp::S4_CppOverloadedMethods ::S4_CppOverloadedMethods
Rcpp::SignedMethod ::is_void
Rcpp::CppMethod::is_void
Rcpp::SignedMethod ::nargs
Rcpp::CppMethod::nargs
Rcpp::SignedMethod ::signature
Rcpp::CppMethod::signature

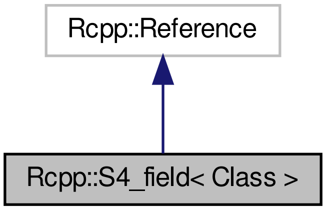
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

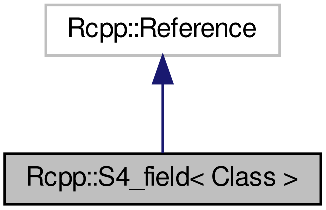
6.704 Rcpp::S4_field< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::S4_field< Class >:



Collaboration diagram for Rcpp::S4_field< Class >:



Public Types

- typedef XPtr< class_Base > XP_Class

Public Member Functions

- S4_field (CppProperty< Class > ∗p, const XP_Class &class_xp)

Private Types

- typedef Rcpp::Reference Base

6.704.1 Detailed Description

template<typename Class>
class Rcpp::S4_field< Class >

Definition at line 321 of file Module.h.

6.704.2 Member Typedef Documentation

6.704.2.1 Base

```
template<typename Class >
typedef Rcpp::Reference Rcpp::S4_field< Class >::Base  [private]
```

Definition at line 322 of file Module.h.

6.704.2.2 XP_Class

```
template<typename Class >
typedef XPtr<class_Base> Rcpp::S4_field< Class >::XP_Class
```

Definition at line 324 of file Module.h.

6.704.3 Constructor & Destructor Documentation

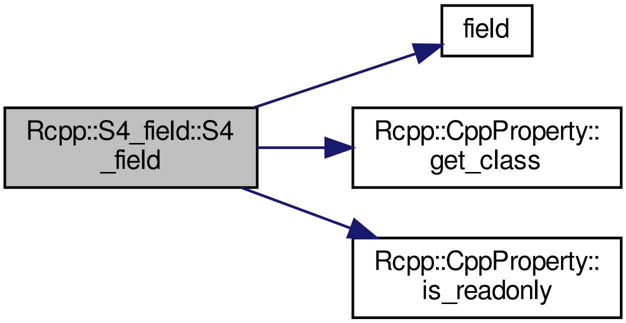
6.704.3.1 S4_field()

```
template<typename Class >
Rcpp::S4_field< Class >::S4_field (
             CppProperty< Class > * p,
             const XP_Class & class_xp )  [inline]
```

Definition at line 325 of file Module.h.

References Rcpp::CppProperty< Class >::docstring, field(), Rcpp::CppProperty< Class >::get_class(), Rcpp::Cpp←Property< Class >::is_readonly(), and RCPP_DEBUG.

Here is the call graph for this function:



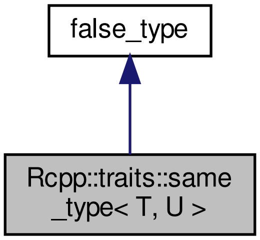
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

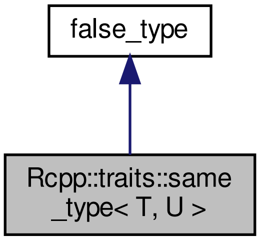
6.705 Rcpp::traits::same_type< T, U > Struct Template Reference

```
#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

6.705.1 Detailed Description

template<typename T, typename U>
struct Rcpp::traits::same_type< T, U >

Definition at line 29 of file same_type.h.

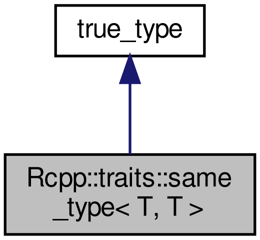
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/same_type.h

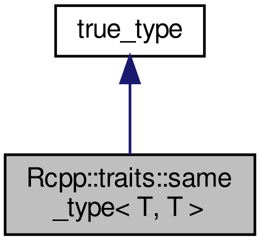
6.706 Rcpp::traits::same_type< T, T > Struct Template Reference

```
#include <same_type.h>
```

Inheritance diagram for Rcpp::traits::same_type< T, T >:



Collaboration diagram for Rcpp::traits::same_type< T, T >:



Additional Inherited Members

6.706.1 Detailed Description

template<typename T>
struct Rcpp::traits::same_type< T, T >

Definition at line 30 of file same_type.h.

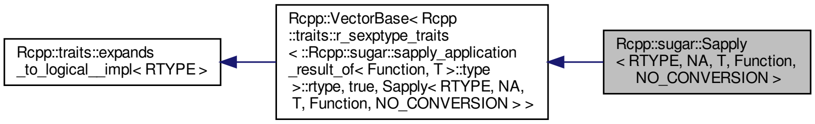
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/same_type.h

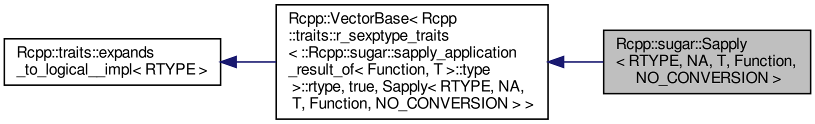
6.707 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION > Class Template Reference

```
#include <sapply.h>
```

Inheritance diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::sugar::Sapply < RTYPE, NA, T, Function, NO_CONVERSION >

Collaboration diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::sugar::Sapply < RTYPE, NA, T, Function, NO_CONVERSION >

Public Types

- typedef ::Rcpp::sugar::sapply_application_result_of< Function, T >::type result_type
- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC
- typedef Rcpp::traits::r_vector_element_converter< RESULT_R_TYPE >::type converter_type
- typedef Rcpp::traits::storage_type< RESULT_R_TYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Sapply (const VEC &vec_, Function fun_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Static Public Attributes

- static const int RESULT_R_TYPE

Private Attributes

- const EXT & vec
- Function fun

6.707.1 Detailed Description

template<int RTYPE, bool NA, typename T, typename Function, bool NO_CONVERSION>
class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >

Definition at line 46 of file sapply.h.

6.707.2 Member Typedef Documentation

6.707.2.1 converter_type

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::r_vector_element_converter<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, NO_CONVERSION >::converter_type
```

Definition at line 59 of file sapply.h.

6.707.2.2 EXT

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
NO_CONVERSION >::EXT
```

Definition at line 62 of file sapply.h.

6.707.2.3 result_type

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef ::Rcpp::sugar::sapply_application_result_of<Function, T>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, NO_CONVERSION >::result_type
```

Definition at line 54 of file sapply.h.

6.707.2.4 STORAGE

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
NO_CONVERSION >::STORAGE
```

Definition at line 60 of file sapply.h.

6.707.2.5 VEC

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION
>::VEC
```

Definition at line 58 of file sapply.h.

6.707.3 Constructor & Destructor Documentation

6.707.3.1 Sapply()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::Sapply (
             const VEC & vec_,
             Function fun_ )  [inline]
```

Definition at line 64 of file sapply.h.

References DEMANGLE, and RCPP_DEBUG_1.

6.707.4 Member Function Documentation

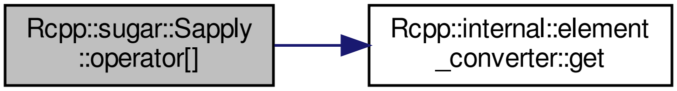
6.707.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
STORAGE Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 69 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::fun, Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec.

Here is the call graph for this function:



6.707.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
R_xlen_t Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::size ( ) const [inline]
```

Definition at line 73 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec.

6.707.5 Member Data Documentation

6.707.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
Function Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::fun [private]
```

Definition at line 77 of file sapply.h.

Referenced by Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[ ](), and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::operator[ ]().

6.707.5.2 RESULT_R_TYPE

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
const int Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::RESULT_R_TYPE  [static]
```

Initial value:

```
=
              Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 55 of file sapply.h.

6.707.5.3 vec

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
const EXT& Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec  [private]
```

Definition at line 76 of file sapply.h.

Referenced by Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[ ](), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::operator[ ](), Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::size(), and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::size().

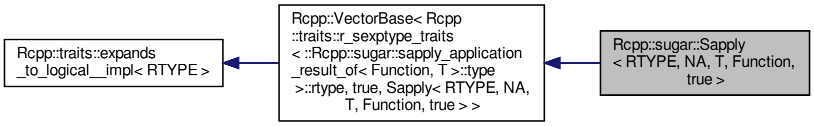
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sapply.h

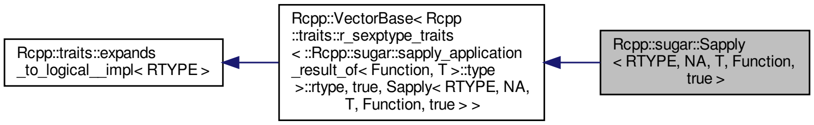
6.708 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true > Class Template Reference

```
#include <sapply.h>
```

Inheritance diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::sugar::Sapply < RTYPE, NA, T, Function, true >

Collaboration diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
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::sugar::Sapply < RTYPE, NA, T, Function, true >

Public Types

- typedef ::Rcpp::sugar::sapply_application_result_of< Function, T >::type result_type
- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC
- typedef Rcpp::traits::storage_type< RESULT_R_TYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Sapply (const VEC &vec_, Function fun_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Static Public Attributes

- static const int RESULT_R_TYPE

Private Attributes

- const EXT & vec
- Function fun

6.708.1 Detailed Description

template<int RTYPE, bool NA, typename T, typename Function>
class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >

Definition at line 83 of file sapply.h.

6.708.2 Member Typedef Documentation

6.708.2.1 EXT

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
true >::EXT
```

Definition at line 98 of file sapply.h.

6.708.2.2 result_type

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef ::Rcpp::sugar::sapply_application_result_of<Function, T>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, true >::result_type
```

Definition at line 91 of file sapply.h.

6.708.2.3 STORAGE

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
true >::STORAGE
```

Definition at line 96 of file sapply.h.

6.708.2.4 VEC

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::VEC
```

Definition at line 95 of file sapply.h.

6.708.3 Constructor & Destructor Documentation

6.708.3.1 Sapply()

```
template<int RTYPE, bool NA, typename T , typename Function >
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::Sapply (
             const VEC & vec_,
             Function fun_ )  [inline]
```

Definition at line 100 of file sapply.h.

References DEMANGLE, and RCPP_DEBUG_1.

6.708.4 Member Function Documentation

6.708.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function >
STORAGE Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 104 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::fun, and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec.

6.708.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function >
R_xlen_t Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::size ( ) const  [inline]
```

Definition at line 107 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec.

6.708.5 Member Data Documentation

6.708.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function >
Function Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::fun  [private]
```

Definition at line 111 of file sapply.h.

6.708.5.2 RESULT_R_TYPE

```
template<int RTYPE, bool NA, typename T , typename Function >
const int Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::RESULT_R_TYPE  [static]
```

Initial value:

```
=
                Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 92 of file sapply.h.

6.708.5.3 vec

```
template<int RTYPE, bool NA, typename T , typename Function >
const EXT& Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::vec  [private]
```

Definition at line 110 of file sapply.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sapply.h

6.709 Rcpp::sugar::sapply_application_result_of< Function, SugarExpression > Struct Template Reference

```
#include <sapply.h>
```

Public Types

- typedef ::Rcpp::traits::result_of< Function >::type type

6.709.1 Detailed Description

template<typename Function, typename SugarExpression>
struct Rcpp::sugar::sapply_application_result_of< Function, SugarExpression >

Definition at line 33 of file sapply.h.

6.709.2 Member Typedef Documentation

6.709.2.1 type

```
template<typename Function , typename SugarExpression >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::sapply_application_result_of<
Function, SugarExpression >::type
```

Definition at line 38 of file sapply.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/sapply.h

6.710 Rcpp::sugar::cbind_impl::scalar< RTYPE > Struct Template Reference

```
#include <cbind.h>
```

Public Types

- typedef cbind_storage_type< RTYPE >::type type

6.710.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::cbind_impl::scalar< RTYPE >

Definition at line 129 of file cbind.h.

6.710.2 Member Typedef Documentation

6.710.2.1 type

```
template<int RTYPE>
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::scalar< RTYPE >::type
```

Definition at line 130 of file cbind.h.

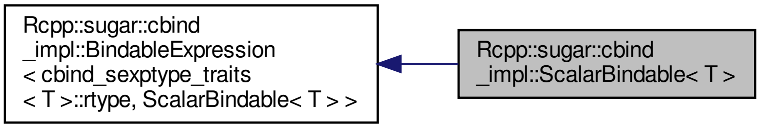
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

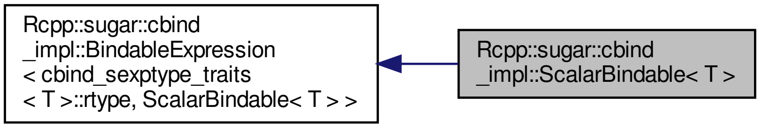
6.711 Rcpp::sugar::cbind_impl::ScalarBindable< T > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind_impl::ScalarBindable< T >:


Rcpp::sugar::cbind _impl::BindableExpression < cbind_sexptype_traits < T >::rtype, ScalarBindable< T > >
Rcpp::sugar::cbind _impl::ScalarBindable< T >

Collaboration diagram for Rcpp::sugar::cbind_impl::ScalarBindable< T >:


Rcpp::sugar::cbind _impl::BindableExpression < cbind_sexptype_traits < T >::rtype, ScalarBindable< T > >
Rcpp::sugar::cbind _impl::ScalarBindable< T >

Public Types

- enum { RTYPE = cbind_sexptype_traits<T>::rtype }
- typedef T stored_type

Public Member Functions

- ScalarBindable (const T &t_)
- R_xlen_t size () const
- R_xlen_t nrow () const
- R_xlen_t ncol () const
- stored_type operator[ ] (R_xlen_t i) const
- stored_type operator() (R_xlen_t i, R_xlen_t j) const

Private Attributes

- T t

6.711.1 Detailed Description

template<typename T>
class Rcpp::sugar::cbind_impl::ScalarBindable< T >

Definition at line 135 of file cbind.h.

6.711.2 Member Typedef Documentation

6.711.2.1 stored_type

```
template<typename T >
typedef T Rcpp::sugar::cbind_impl::ScalarBindable< T >::stored_type
```

Definition at line 139 of file cbind.h.

6.711.3 Member Enumeration Documentation

6.711.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	

Definition at line 140 of file cbind.h.

6.711.4 Constructor & Destructor Documentation

6.711.4.1 ScalarBindable()

```
template<typename T >
Rcpp::sugar::cbind_impl::ScalarBindable< T >::ScalarBindable (
            const T & t_ )  [inline]
```

Definition at line 146 of file cbind.h.

6.711.5 Member Function Documentation

6.711.5.1 ncol()

```
template<typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::ncol ( ) const  [inline]
```

Definition at line 152 of file cbind.h.

6.711.5.2 nrow()

```
template<typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::nrow ( ) const  [inline]
```

Definition at line 150 of file cbind.h.

6.711.5.3 operator()()

```
template<typename T >
stored_type Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator() (
             R_xlen_t i,
             R_xlen_t j ) const  [inline]
```

Definition at line 158 of file cbind.h.

References Rcpp::sugar::cbind_impl::ScalarBindable< T >::t.

6.711.5.4 operator[]()

```
template<typename T >
stored_type Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 154 of file cbind.h.

References Rcpp::sugar::cbind_impl::ScalarBindable< T >::t.

6.711.5.5 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::size ( ) const  [inline]
```

Definition at line 148 of file cbind.h.

6.711.6 Member Data Documentation

6.711.6.1 t

```
template<typename T >
T Rcpp::sugar::cbind_impl::ScalarBindable< T >::t  [private]
```

Definition at line 143 of file cbind.h.

Referenced by Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator()(), and Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator[ ]().

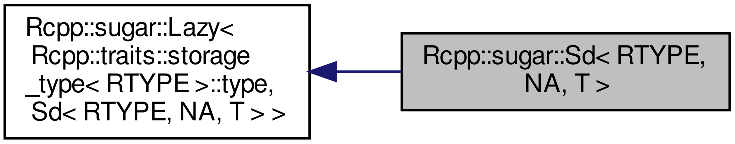
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

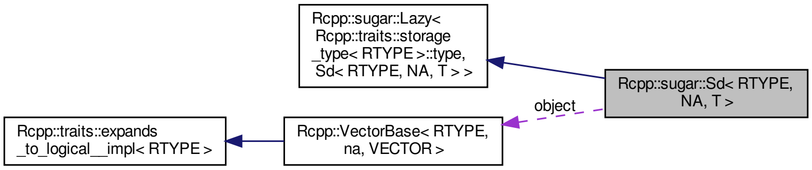
6.712 Rcpp::sugar::Sd< RTYPE, NA, T > Class Template Reference

```
#include <sd.h>
```

Inheritance diagram for Rcpp::sugar::Sd< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sd< RTYPE, NA, T > >
Rcpp::sugar::Sd< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Sd< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sd< RTYPE, NA, T > >
Rcpp::sugar::Sd< RTYPE, NA, T >
object_
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Sd (const VEC_TYPE &object_)
- STORAGE get () const

Private Attributes

- const VEC_TYPE & object

6.712.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sd< RTYPE, NA, T >

Definition at line 29 of file sd.h.

6.712.2 Member Typedef Documentation

6.712.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sd< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file sd.h.

6.712.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sd< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file sd.h.

6.712.3 Constructor & Destructor Documentation

6.712.3.1 Sd()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sd< RTYPE, NA, T >::Sd (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 34 of file sd.h.

6.712.4 Member Function Documentation

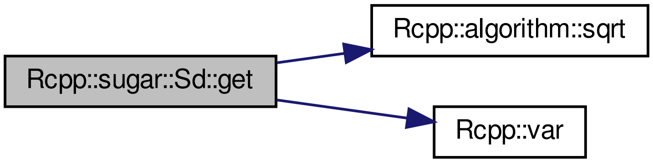
6.712.4.1 get()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Sd< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 36 of file sd.h.

References Rcpp::algorithm::sqrt(), and Rcpp::var().

Here is the call graph for this function:



6.712.5 Member Data Documentation

6.712.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Sd< RTYPE, NA, T >::object  [private]
```

Definition at line 40 of file sd.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sd.h

6.713 Rcpp::sugar::SelfHash< RTYPE > Class Template Reference

```
#include <SelfHash.h>
```

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Vector< RTYPE > VECTOR

Public Member Functions

- SelfHash (SEXP table)
- IntegerVector fill_and_self_match ()
- int size () const
- int add_value_get_index (int i)
- unsigned int get_index (STORAGE value) const
- unsigned int get_addr (STORAGE value) const
- unsigned int get_addr (int value) const
- unsigned int get_addr (double val) const
- unsigned int get_addr (SEXP value) const

Public Attributes

- int n
- int m
- int k
- STORAGE ∗ src
- std::vector< int > data
- std::vector< int > indices
- int size_

6.713.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::SelfHash< RTYPE >

Definition at line 32 of file SelfHash.h.

6.713.2 Member Typedef Documentation

6.713.2.1 STORAGE

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::SelfHash< RTYPE >::STORAGE
```

Definition at line 34 of file SelfHash.h.

6.713.2.2 VECTOR

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::SelfHash< RTYPE >::VECTOR
```

Definition at line 35 of file SelfHash.h.

6.713.3 Constructor & Destructor Documentation

6.713.3.1 SelfHash()

```
template<int RTYPE>
Rcpp::sugar::SelfHash< RTYPE >::SelfHash (
             SEXP table )  [inline]
```

Definition at line 37 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::data, Rcpp::sugar::SelfHash< RTYPE >::indices, Rcpp::sugar::SelfHash< RTYPE >::k, Rcpp::sugar::SelfHash< RTYPE >::m, and Rcpp::sugar::SelfHash< RTYPE >::n.

6.713.4 Member Function Documentation

6.713.4.1 add_value_get_index()

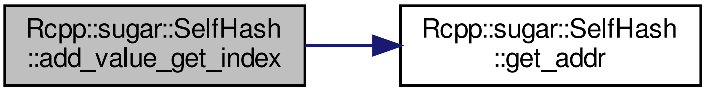
```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index (
            int i )  [inline]
```

Definition at line 63 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::data, Rcpp::sugar::SelfHash< RTYPE >::get_addr(), Rcpp::sugar::↩ SelfHash< RTYPE >::indices, Rcpp::sugar::SelfHash< RTYPE >::m, Rcpp::sugar::SelfHash< RTYPE >::size_, and Rcpp::sugar::SelfHash< RTYPE >::src.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match().

Here is the call graph for this function:


Rcpp::sugar::SelfHash ::add_value_get_index
Rcpp::sugar::SelfHash ::get_addr

6.713.4.2 fill_and_self_match()

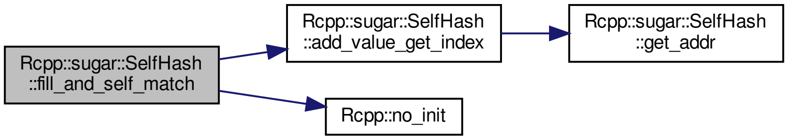
```
template<int RTYPE>
IntegerVector Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match ( )  [inline]
```

Definition at line 46 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), Rcpp::sugar::SelfHash< RTYPE >::n, and Rcpp::no_init().

Referenced by Rcpp::self_match().

Here is the call graph for this function:


Rcpp::sugar::SelfHash ::fill_and_self_match
Rcpp::sugar::SelfHash ::add_value_get_index
Rcpp::sugar::SelfHash ::get_addr
Rcpp::no_init

6.713.4.3 get_addr() [1/4]

```
unsigned int Rcpp::sugar::SelfHash< REALSXP >::get_addr (
             double val ) const  [inline]
```

Definition at line 98 of file SelfHash.h.

References RCPP_HASH.

6.713.4.4 get_addr() [2/4]

```
unsigned int Rcpp::sugar::SelfHash< INTSXP >::get_addr (
             int value ) const  [inline]
```

Definition at line 94 of file SelfHash.h.

References RCPP_HASH.

6.713.4.5 get_addr() [3/4]

```
unsigned int Rcpp::sugar::SelfHash< STRSXP >::get_addr (
             SEXP value ) const  [inline]
```

Definition at line 115 of file SelfHash.h.

References RCPP_HASH.

6.713.4.6 get_addr() [4/4]

```
template<int RTYPE>
unsigned int Rcpp::sugar::SelfHash< RTYPE >::get_addr (
             STORAGE value ) const
```

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), and Rcpp::sugar::SelfHash< RTYPE >↩ ::get_index().

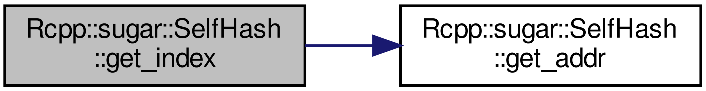
6.713.4.7 get_index()

```
template<int RTYPE>
unsigned int Rcpp::sugar::SelfHash< RTYPE >::get_index (
             STORAGE value ) const  [inline]
```

Definition at line 78 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::data, Rcpp::sugar::SelfHash< RTYPE >::get_addr(), Rcpp::sugar::↩ SelfHash< RTYPE >::m, and Rcpp::sugar::SelfHash< RTYPE >::src.

Here is the call graph for this function:


Rcpp::sugar::SelfHash ::get_index
Rcpp::sugar::SelfHash ::get_addr

6.713.4.8 size()

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::size ( ) const  [inline]
```

Definition at line 53 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::size_.

6.713.5 Member Data Documentation

6.713.5.1 data

```
template<int RTYPE>
std::vector<int> Rcpp::sugar::SelfHash< RTYPE >::data
```

Definition at line 59 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), Rcpp::sugar::SelfHash< RTYPE >::get_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.713.5.2 indices

```
template<int RTYPE>
std::vector<int> Rcpp::sugar::SelfHash< RTYPE >::indices
```

Definition at line 60 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.713.5.3 k

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::k
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.713.5.4 m

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::m
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), Rcpp::sugar::SelfHash< RTYPE >::get_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

6.713.5.5 n

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::n
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match(), and Rcpp::sugar::SelfHash< RTYPE >::↩ SelfHash().

6.713.5.6 size_

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::size_
```

Definition at line 61 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), and Rcpp::sugar::SelfHash< RTYPE >↩ ::size().

6.713.5.7 src

```
template<int RTYPE>
STORAGE* Rcpp::sugar::SelfHash< RTYPE >::src
```

Definition at line 58 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index(), and Rcpp::sugar::SelfHash< RTYPE >↩ ::get_index().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/hash/SelfHash.h

6.714 Rcpp::sugar::SelfInserter< HASH, STORAGE > Class Template Reference

```
#include <self_match.h>
```

Public Member Functions

- SelfInserter (HASH &hash_)
- R_xlen_t operator() (STORAGE value)

Private Attributes

- HASH & hash
- R_xlen_t index

6.714.1 Detailed Description

template<typename HASH, typename STORAGE>
class Rcpp::sugar::SelfInserter< HASH, STORAGE >

Definition at line 29 of file self_match.h.

6.714.2 Constructor & Destructor Documentation

6.714.2.1 SelfInserter()

```
template<typename HASH , typename STORAGE >
Rcpp::sugar::SelfInserter< HASH, STORAGE >::SelfInserter (
             HASH & hash_ )  [inline]
```

Definition at line 31 of file self_match.h.

6.714.3 Member Function Documentation

6.714.3.1 operator()()

```
template<typename HASH , typename STORAGE >
R_xlen_t Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator() (
             STORAGE value )  [inline]
```

Definition at line 33 of file self_match.h.

References Rcpp::sugar::SelfInserter< HASH, STORAGE >::hash, and Rcpp::sugar::SelfInserter< HASH, STORAGE >::index.

6.714.4 Member Data Documentation

6.714.4.1 hash

```
template<typename HASH , typename STORAGE >
HASH& Rcpp::sugar::SelfInserter< HASH, STORAGE >::hash  [private]
```

Definition at line 44 of file self_match.h.

Referenced by Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator()().

6.714.4.2 index

```
template<typename HASH , typename STORAGE >
R_xlen_t Rcpp::sugar::SelfInserter< HASH, STORAGE >::index  [private]
```

Definition at line 45 of file self_match.h.

Referenced by Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator()().

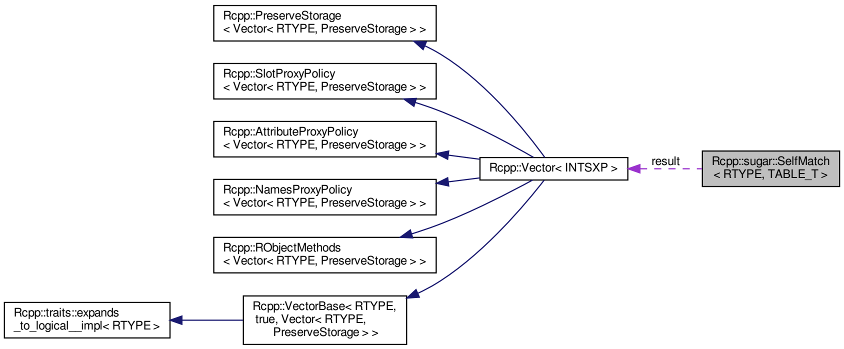
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self_match.h

6.715 Rcpp::sugar::SelfMatch< RTYPE, TABLE_T > Class Template Reference

```
#include <self_match.h>
```

Collaboration diagram for Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >:



Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- SelfMatch (const TABLE_T &table)
- operator IntegerVector () const

Private Types

- typedef RCPP_UNORDERED_MAP< STORAGE, int > HASH
- typedef SelfInserter< HASH, STORAGE > Inserter

Private Attributes

- HASH hash
- IntegerVector result

6.715.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >

Definition at line 49 of file self_match.h.

6.715.2 Member Typedef Documentation

6.715.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef RCPP_UNORDERED_MAP<STORAGE, int> Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::HASH  [private]
```

Definition at line 60 of file self_match.h.

6.715.2.2 Inserter

```
template<int RTYPE, typename TABLE_T >
typedef SelfInserter<HASH,STORAGE> Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::Inserter  [private]
```

Definition at line 61 of file self_match.h.

6.715.2.3 STORAGE

```
template<int RTYPE, typename TABLE_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::STORAGE
```

Definition at line 51 of file self_match.h.

6.715.3 Constructor & Destructor Documentation

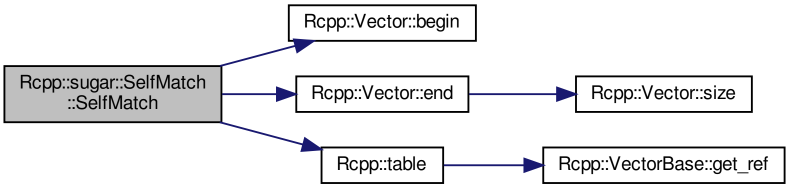
6.715.3.1 SelfMatch()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch (
             const TABLE_T & table )  [inline]
```

Definition at line 53 of file self_match.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::hash, and Rcpp::table().

Here is the call graph for this function:



6.715.4 Member Function Documentation

6.715.4.1 operator IntegerVector()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::operator IntegerVector ( ) const  [inline]
```

Definition at line 57 of file self_match.h.

References Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::result.

6.715.5 Member Data Documentation

6.715.5.1 hash

```
template<int RTYPE, typename TABLE_T >
HASH Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::hash  [private]
```

Definition at line 62 of file self_match.h.

Referenced by Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch().

6.715.5.2 result

```
template<int RTYPE, typename TABLE_T >
IntegerVector Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::result  [private]
```

Definition at line 63 of file self_match.h.

Referenced by Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::operator IntegerVector().

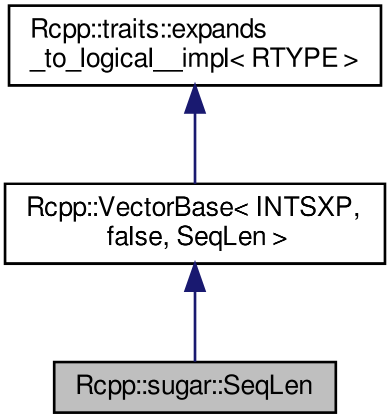
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self_match.h

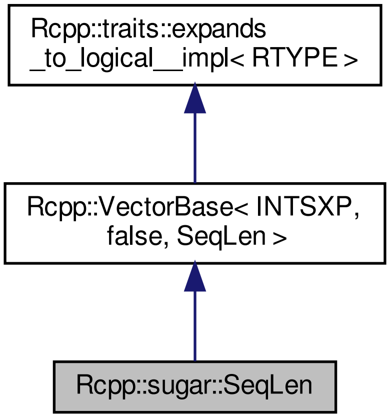
6.716 Rcpp::sugar::SeqLen Class Reference

```
#include <seq_along.h>
```

Inheritance diagram for Rcpp::sugar::SeqLen:



Collaboration diagram for Rcpp::sugar::SeqLen:



Public Member Functions

- SeqLen (R_xlen_t len_)
- R_xlen_t operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- R_xlen_t len

Additional Inherited Members

6.716.1 Detailed Description

Definition at line 28 of file seq_along.h.

6.716.2 Constructor & Destructor Documentation

6.716.2.1 SeqLen()

```
Rcpp::sugar::SeqLen::SeqLen (
            R_xlen_t len_ )  [inline]
```

Definition at line 30 of file seq_along.h.

6.716.3 Member Function Documentation

6.716.3.1 operator[]()

```
R_xlen_t Rcpp::sugar::SeqLen::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 32 of file seq_along.h.

6.716.3.2 size()

```
R_xlen_t Rcpp::sugar::SeqLen::size ( ) const  [inline]
```

Definition at line 35 of file seq_along.h.

References len.

6.716.4 Member Data Documentation

6.716.4.1 len

```
R_xlen_t Rcpp::sugar::SeqLen::len  [private]
```

Definition at line 38 of file seq_along.h.

Referenced by size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/seq_along.h

6.717 Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <setdiff.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- SetDiff (const LHS_T &lhs, const RHS_T &rhs)
- Vector< RTYPE > get () const

Private Types

- typedef RCPP_UNORDERED_SET< STORAGE > SET
- typedef SET::const_iterator ITERATOR

Private Attributes

- SET lhs_set
- SET rhs_set

6.717.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 43 of file setdiff.h.

6.717.2 Member Typedef Documentation

6.717.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 64 of file setdiff.h.

6.717.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_↩
T >::SET  [private]
```

Definition at line 63 of file setdiff.h.

6.717.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS↩
_NA, RHS_T >::STORAGE
```

Definition at line 45 of file setdiff.h.

6.717.3 Constructor & Destructor Documentation

6.717.3.1 SetDiff()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetDiff (
            const LHS_T & lhs,
            const RHS_T & rhs )  [inline]
```

Definition at line 47 of file setdiff.h.

References Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set, and Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set.

6.717.4 Member Function Documentation

6.717.4.1 get()

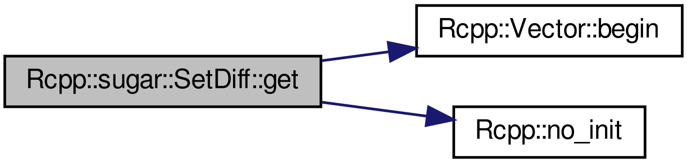
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const  [inline]
```

Definition at line 55 of file setdiff.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set, and Rcpp::no_init().

Referenced by Rcpp::setdiff().

Here is the call graph for this function:



6.717.5 Member Data Documentation

6.717.5.1 lhs_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set  [private]
```

Definition at line 65 of file setdiff.h.

Referenced by Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), and Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetDiff().

6.717.5.2 rhs_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set  [private]
```

Definition at line 66 of file setdiff.h.

Referenced by Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetDiff().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

6.718 Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <setdiff.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- SetEqual (const LHS_T &lhs, const RHS_T &rhs)
- bool get () const

Private Types

- typedef RCPP_UNORDERED_SET< STORAGE > SET
- typedef SET::const_iterator ITERATOR

Private Attributes

- SET lhs_set
- SET rhs_set

6.718.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 71 of file setdiff.h.

6.718.2 Member Typedef Documentation

6.718.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 93 of file setdiff.h.

6.718.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_↩
T >::SET  [private]
```

Definition at line 92 of file setdiff.h.

6.718.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 73 of file setdiff.h.

6.718.3 Constructor & Destructor Documentation

6.718.3.1 SetEqual()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetEqual (
            const LHS_T & lhs,
            const RHS_T & rhs )  [inline]
```

Definition at line 75 of file setdiff.h.

6.718.4 Member Function Documentation

6.718.4.1 get()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
bool Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const  [inline]
```

Definition at line 81 of file setdiff.h.

References Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set, and Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set.

Referenced by Rcpp::setequal().

6.718.5 Member Data Documentation

6.718.5.1 lhs_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set  [private]
```

Definition at line 94 of file setdiff.h.

Referenced by Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get().

6.718.5.2 rhs_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set  [private]
```

Definition at line 95 of file setdiff.h.

Referenced by Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

6.719 Rcpp::Shelter< T > Class Template Reference

```
#include <Shelter.h>
```

Public Member Functions

- Shelter ()
- SEXP operator() (SEXP x)
- ~Shelter ()

Private Member Functions

- Shelter (const Shelter &)
- Shelter & operator= (const Shelter &)

Private Attributes

- int nprotected

6.719.1 Detailed Description

template<class T>
class Rcpp::Shelter< T >

Definition at line 24 of file Shelter.h.

6.719.2 Constructor & Destructor Documentation

6.719.2.1 Shelter() [1/2]

```
template<class T >
Rcpp::Shelter< T >::Shelter ( )  [inline]
```

Definition at line 26 of file Shelter.h.

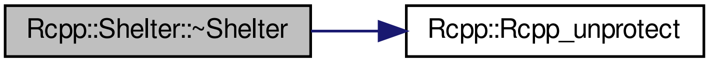
6.719.2.2 ∼Shelter()

```
template<class T >
Rcpp::Shelter< T >::∼Shelter ( )  [inline]
```

Definition at line 33 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected, and Rcpp::Rcpp_unprotect().

Here is the call graph for this function:



6.719.2.3 Shelter() [2/2]

```
template<class T >
Rcpp::Shelter< T >::Shelter (
             const Shelter< T > & ) [private]
```

6.719.3 Member Function Documentation

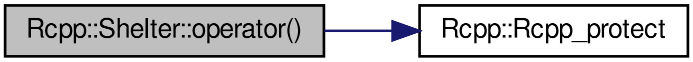
6.719.3.1 operator()()

```
template<class T >
SEXP Rcpp::Shelter< T >::operator() (
             SEXP x ) [inline]
```

Definition at line 28 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected, and Rcpp::Rcpp_protect().

Here is the call graph for this function:



6.719.3.2 operator=()

```
template<class T >
Shelter& Rcpp::Shelter< T >::operator= (
             const Shelter< T > & ) [private]
```

6.719.4 Member Data Documentation

6.719.4.1 nprotected

```
template<class T >
int Rcpp::Shelter< T >::nprotected  [private]
```

Definition at line 39 of file Shelter.h.

Referenced by Rcpp::Shelter< T >::operator()(), and Rcpp::Shelter< T >::∼Shelter().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/protection/Shelter.h

6.720 Rcpp::Shield< T > Class Template Reference

```
#include <Shield.h>
```

Public Member Functions

- Shield (SEXP t_)
- ∼Shield ()
- operator SEXP () const

Public Attributes

- SEXP t

Private Member Functions

- Shield (const Shield &)
- Shield & operator= (const Shield &)

6.720.1 Detailed Description

template<typename T>
class Rcpp::Shield< T >

Definition at line 35 of file Shield.h.

6.720.2 Constructor & Destructor Documentation

6.720.2.1 Shield() [1/2]

```
template<typename T >
Rcpp::Shield< T >::Shield (
             SEXP t_ )  [inline]
```

Definition at line 37 of file Shield.h.

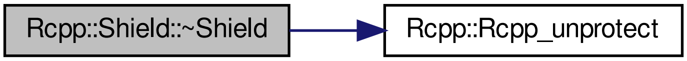
6.720.2.2 ∼Shield()

```
template<typename T >
Rcpp::Shield< T >::∼Shield ( )  [inline]
```

Definition at line 38 of file Shield.h.

References Rcpp::Rcpp_unprotect(), and Rcpp::Shield< T >::t.

Here is the call graph for this function:



6.720.2.3 Shield() [2/2]

```
template<typename T >
Rcpp::Shield< T >::Shield (
             const Shield< T > &  )  [private]
```

6.720.3 Member Function Documentation

6.720.3.1 operator SEXP()

```
template<typename T >
Rcpp::Shield< T >::operator SEXP ( ) const  [inline]
```

Definition at line 42 of file Shield.h.

References Rcpp::Shield< T >::t.

6.720.3.2 operator=()

```
template<typename T >
Shield& Rcpp::Shield< T >::operator= (
             const Shield< T > &  )  [private]
```

6.720.4 Member Data Documentation

6.720.4.1 t

```
template<typename T >
SEXP Rcpp::Shield< T >::t
```

Definition at line 43 of file Shield.h.

Referenced by Rcpp::Shield< T >::operator SEXP(), and Rcpp::Shield< T >::~Shield().

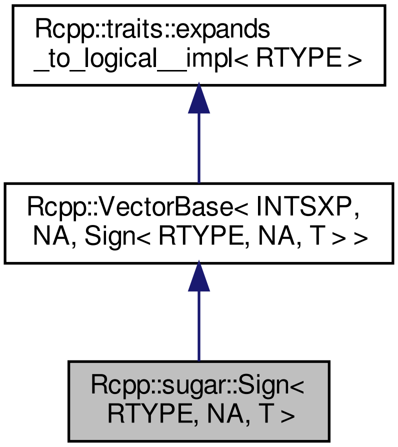
The documentation for this class was generated from the following file:

- inst/include/Rcpp/protection/Shield.h

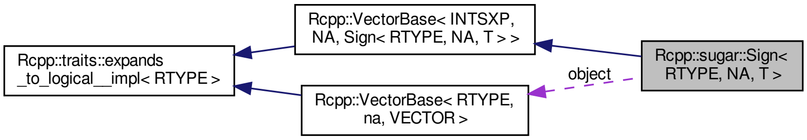
6.721 Rcpp::sugar::Sign< RTYPE, NA, T > Class Template Reference

```
#include <sign.h>
```

Inheritance diagram for Rcpp::sugar::Sign< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > >
Rcpp::sugar::Sign< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Sign< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > >
Rcpp::VectorBase< RTYPE, na, VECTOR >
object
Rcpp::sugar::Sign< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef int r_import_type

Public Member Functions

- Sign (const VEC_TYPE &object_)
- int operator[ ] (R_xlen_t i) const
- R_xlen_t size () const
- operator SEXP () const
- int get (R_xlen_t i) const

Private Attributes

- const VEC_TYPE & object

6.721.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sign< RTYPE, NA, T >

Definition at line 48 of file sign.h.

6.721.2 Member Typedef Documentation

6.721.2.1 r_import_type

```
template<int RTYPE, bool NA, typename T >
typedef int Rcpp::sugar::Sign< RTYPE, NA, T >::r_import_type
```

Definition at line 52 of file sign.h.

6.721.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sign< RTYPE, NA, T >::STORAGE
```

Definition at line 51 of file sign.h.

6.721.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sign< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 50 of file sign.h.

6.721.3 Constructor & Destructor Documentation

6.721.3.1 Sign()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sign< RTYPE, NA, T >::Sign (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 54 of file sign.h.

6.721.4 Member Function Documentation

6.721.4.1 get()

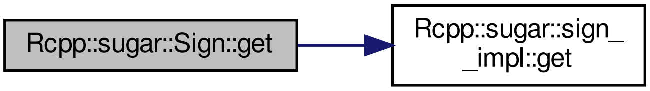
```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Sign< RTYPE, NA, T >::get (
             R_xlen_t i ) const  [inline]
```

Definition at line 62 of file sign.h.

References Rcpp::sugar::sign__impl< NA, RTYPE >::get().

Referenced by Rcpp::sugar::Sign< RTYPE, NA, T >::operator[ ]().

Here is the call graph for this function:



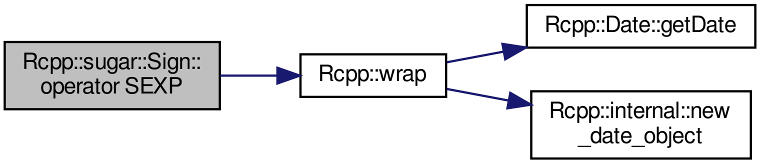
6.721.4.2 operator SEXP()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP ( ) const  [inline]
```

Definition at line 61 of file sign.h.

References Rcpp::wrap().

Here is the call graph for this function:



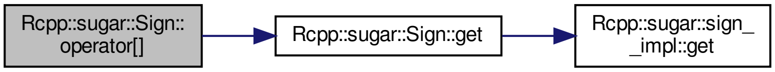
6.721.4.3 operator[]()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Sign< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 56 of file sign.h.

References Rcpp::sugar::Sign< RTYPE, NA, T >::get().

Here is the call graph for this function:



6.721.4.4 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Sign< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 59 of file sign.h.

6.721.5 Member Data Documentation

6.721.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Sign< RTYPE, NA, T >::object  [private]
```

Definition at line 64 of file sign.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sign.h

6.722 Rcpp::sugar::sign__impl< NA, RTYPE > Class Template Reference

```
#include <sign.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Static Public Member Functions

- static int get (STORAGE x)

6.722.1 Detailed Description

**template<bool NA, int RTYPE>
class Rcpp::sugar::sign__impl< NA, RTYPE >**

Definition at line 29 of file sign.h.

6.722.2 Member Typedef Documentation

6.722.2.1 STORAGE

```
template<bool NA, int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::sign__impl< NA, RTYPE >::STORAGE
```

Definition at line 31 of file sign.h.

6.722.3 Member Function Documentation

6.722.3.1 get()

```
template<bool NA, int RTYPE>
static int Rcpp::sugar::sign__impl< NA, RTYPE >::get (
             STORAGE x )  [inline], [static]
```

Definition at line 32 of file sign.h.

Referenced by Rcpp::sugar::Sign< RTYPE, NA, T >::get().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sign.h

6.723 Rcpp::sugar::sign__impl< false, RTYPE > Class Template Reference

```
#include <sign.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Static Public Member Functions

- static int get (STORAGE x)

6.723.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::sign__impl< false, RTYPE >

Definition at line 38 of file sign.h.

6.723.2 Member Typedef Documentation

6.723.2.1 STORAGE

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::sign__impl< false, RTYPE >::STORAGE
```

Definition at line 40 of file sign.h.

6.723.3 Member Function Documentation

6.723.3.1 get()

```
template<int RTYPE>
static int Rcpp::sugar::sign__impl< false, RTYPE >::get (
             STORAGE x )  [inline], [static]
```

Definition at line 41 of file sign.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sign.h

6.724 Rcpp::SignedConstructor< Class > Class Template Reference

```
#include <Module.h>
```

Public Member Functions

- SignedConstructor (Constructor_Base< Class > ∗ctor_, ValidConstructor valid_, const char ∗doc)
- int nargs ()
- void signature (std::string &buffer, const std::string &class_name)

Public Attributes

- Constructor_Base< Class > * ctor
- ValidConstructor valid
- std::string docstring

6.724.1 Detailed Description

template<typename Class>
class Rcpp::SignedConstructor< Class >

Definition at line 143 of file Module.h.

6.724.2 Constructor & Destructor Documentation

6.724.2.1 SignedConstructor()

```
template<typename Class >
Rcpp::SignedConstructor< Class >::SignedConstructor (
             Constructor_Base< Class > * ctor_,
             ValidConstructor valid_,
             const char * doc )  [inline]
```

Definition at line 146 of file Module.h.

6.724.3 Member Function Documentation

6.724.3.1 nargs()

```
template<typename Class >
int Rcpp::SignedConstructor< Class >::nargs ( )  [inline]
```

Definition at line 156 of file Module.h.

References Rcpp::SignedConstructor< Class >::ctor.

Referenced by Rcpp::S4_CppConstructor< Class >::S4_CppConstructor().

6.724.3.2 signature()

```
template<typename Class >
void Rcpp::SignedConstructor< Class >::signature (
            std::string & buffer,
            const std::string & class_name )  [inline]
```

Definition at line 157 of file Module.h.

References Rcpp::SignedConstructor< Class >::ctor.

Referenced by Rcpp::S4_CppConstructor< Class >::S4_CppConstructor().

6.724.4 Member Data Documentation

6.724.4.1 ctor

```
template<typename Class >
Constructor_Base<Class>* Rcpp::SignedConstructor< Class >::ctor
```

Definition at line 152 of file Module.h.

Referenced by Rcpp::SignedConstructor< Class >::nargs(), and Rcpp::SignedConstructor< Class >::signature().

6.724.4.2 docstring

```
template<typename Class >
std::string Rcpp::SignedConstructor< Class >::docstring
```

Definition at line 154 of file Module.h.

Referenced by Rcpp::S4_CppConstructor< Class >::S4_CppConstructor().

6.724.4.3 valid

```
template<typename Class >
ValidConstructor Rcpp::SignedConstructor< Class >::valid
```

Definition at line 153 of file Module.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.725 Rcpp::SignedFactory< Class > Class Template Reference

```
#include <Module.h>
```

Public Member Functions

- SignedFactory (Factory_Base< Class > ∗fact_, ValidConstructor valid_, const char ∗doc)
- int nargs ()
- void signature (std::string &buffer, const std::string &class_name)

Public Attributes

- Factory_Base< Class > ∗ fact
- ValidConstructor valid
- std::string docstring

6.725.1 Detailed Description

template<typename Class>
class Rcpp::SignedFactory< Class >

Definition at line 163 of file Module.h.

6.725.2 Constructor & Destructor Documentation

6.725.2.1 SignedFactory()

```
template<typename Class >
Rcpp::SignedFactory< Class >::SignedFactory (
            Factory_Base< Class > ∗ fact_,
            ValidConstructor valid_,
            const char ∗ doc ) [inline]
```

Definition at line 166 of file Module.h.

6.725.3 Member Function Documentation

6.725.3.1 nargs()

```
template<typename Class >
int Rcpp::SignedFactory< Class >::nargs ( ) [inline]
```

Definition at line 176 of file Module.h.

References Rcpp::SignedFactory< Class >::fact.

6.725.3.2 signature()

```
template<typename Class >
void Rcpp::SignedFactory< Class >::signature (
             std::string & buffer,
             const std::string & class_name ) [inline]
```

Definition at line 177 of file Module.h.

References Rcpp::SignedFactory< Class >::fact.

6.725.4 Member Data Documentation

6.725.4.1 docstring

```
template<typename Class >
std::string Rcpp::SignedFactory< Class >::docstring
```

Definition at line 174 of file Module.h.

6.725.4.2 fact

```
template<typename Class >
Factory_Base<Class>* Rcpp::SignedFactory< Class >::fact
```

Definition at line 172 of file Module.h.

Referenced by Rcpp::SignedFactory< Class >::nargs(), and Rcpp::SignedFactory< Class >::signature().

6.725.4.3 valid

```
template<typename Class >
ValidConstructor Rcpp::SignedFactory< Class >::valid
```

Definition at line 173 of file Module.h.

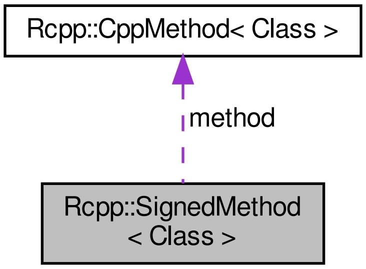
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

6.726 Rcpp::SignedMethod< 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

6.726.1 Detailed Description

template<typename Class>
class Rcpp::SignedMethod< Class >

Definition at line 184 of file Module.h.

6.726.2 Member Typedef Documentation

6.726.2.1 METHOD

```
template<typename Class >
typedef CppMethod<Class> Rcpp::SignedMethod< Class >::METHOD
```

Definition at line 186 of file Module.h.

6.726.3 Constructor & Destructor Documentation

6.726.3.1 SignedMethod()

```
template<typename Class >
Rcpp::SignedMethod< Class >::SignedMethod (
             METHOD ∗ m,
             ValidMethod valid_,
             const char ∗ doc )  [inline]
```

Definition at line 187 of file Module.h.

6.726.4 Member Function Documentation

6.726.4.1 is_const()

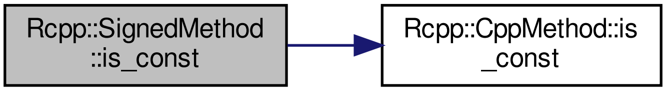
```
template<typename Class >
bool Rcpp::SignedMethod< Class >::is_const ( ) [inline]
```

Definition at line 195 of file Module.h.

References Rcpp::CppMethod< Class >::is_const(), and Rcpp::SignedMethod< Class >::method.

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



6.726.4.2 is_void()

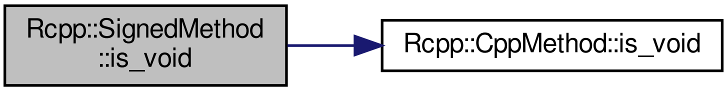
```
template<typename Class >
bool Rcpp::SignedMethod< Class >::is_void ( ) [inline]
```

Definition at line 194 of file Module.h.

References Rcpp::CppMethod< Class >::is_void(), and Rcpp::SignedMethod< Class >::method.

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



6.726.4.3 nargs()

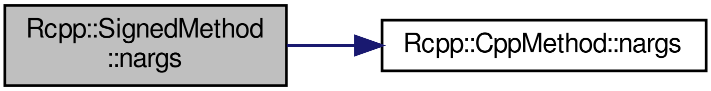
```
template<typename Class >
int Rcpp::SignedMethod< Class >::nargs ( )  [inline]
```

Definition at line 193 of file Module.h.

References Rcpp::SignedMethod< Class >::method, and Rcpp::CppMethod< Class >::nargs().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



6.726.4.4 signature()

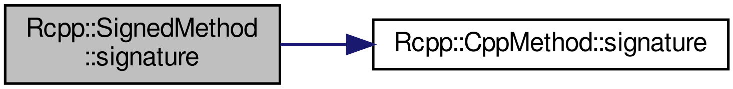
```
template<typename Class >
void Rcpp::SignedMethod< Class >::signature (
             std::string & s,
             const char * name )  [inline]
```

Definition at line 196 of file Module.h.

References Rcpp::SignedMethod< Class >::method, and Rcpp::CppMethod< Class >::signature().

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

Here is the call graph for this function:



6.726.5 Member Data Documentation

6.726.5.1 docstring

```
template<typename Class >
std::string Rcpp::SignedMethod< Class >::docstring
```

Definition at line 191 of file Module.h.

Referenced by Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods().

6.726.5.2 method

```
template<typename Class >
METHOD* Rcpp::SignedMethod< Class >::method
```

Definition at line 189 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::is_const(), Rcpp::SignedMethod< Class >::is_void(), Rcpp::SignedMethod< Class >::nargs(), and Rcpp::SignedMethod< Class >::signature().

6.726.5.3 valid

```
template<typename Class >
ValidMethod Rcpp::SignedMethod< Class >::valid
```

Definition at line 190 of file Module.h.

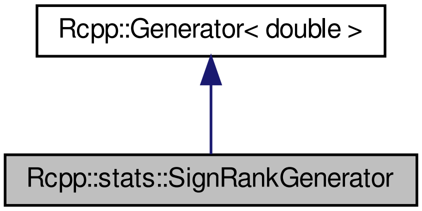
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

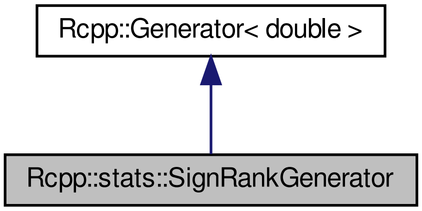
6.727 Rcpp::stats::SignRankGenerator Class Reference

```
#include <rsignrank.h>
```

Inheritance diagram for Rcpp::stats::SignRankGenerator:



Collaboration diagram for Rcpp::stats::SignRankGenerator:



Public Member Functions

- SignRankGenerator (double nn_)
- double operator() () const

Private Attributes

- double nn

Additional Inherited Members

6.727.1 Detailed Description

Definition at line 29 of file rsignrank.h.

6.727.2 Constructor & Destructor Documentation

6.727.2.1 SignRankGenerator()

```
Rcpp::stats::SignRankGenerator::SignRankGenerator (
             double nn_ )  [inline]
```

Definition at line 31 of file rsignrank.h.

6.727.3 Member Function Documentation

6.727.3.1 operator()()

```
double Rcpp::stats::SignRankGenerator::operator() ( ) const  [inline]
```

Definition at line 32 of file rsignrank.h.

References nn.

6.727.4 Member Data Documentation

6.727.4.1 nn

```
double Rcpp::stats::SignRankGenerator::nn  [private]
```

Definition at line 34 of file rsignrank.h.

Referenced by operator()().

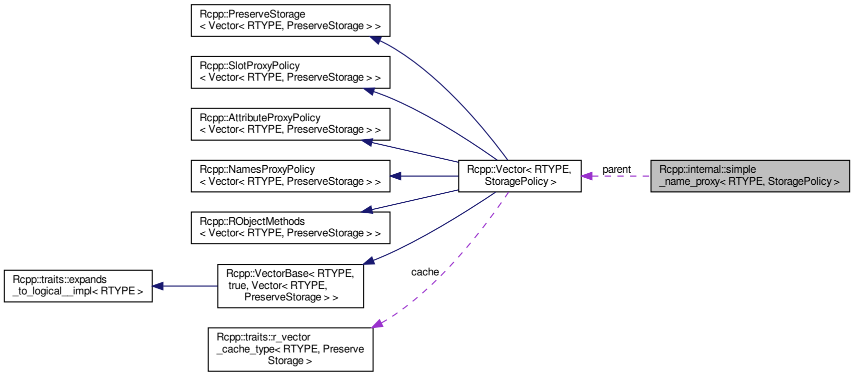
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rsignrank.h

6.728 Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, StoragePolicy >
parent
Rcpp::internal::simple _name_proxy< RTYPE, StoragePolicy >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
cache
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef ::Rcpp::traits::storage_type< RTYPE >::type CTYPE

Public Member Functions

- simple_name_proxy (VECTOR &v, const std::string &name_)
- simple_name_proxy (const simple_name_proxy &other)
- ∼simple_name_proxy ()
- simple_name_proxy & operator= (CTYPE rhs)
- simple_name_proxy & operator= (const simple_name_proxy &other)
- template<typename T >
  simple_name_proxy & operator= (const T &rhs)
- operator CTYPE () const
- operator SEXP () const

Private Member Functions

- void set (CTYPE rhs)
- CTYPE get () const

Private Attributes

- VECTOR & parent
- std::string name

6.728.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >

Definition at line 29 of file proxy.h.

6.728.2 Member Typedef Documentation

6.728.2.1 CTYPE

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::simple_name_proxy< RTYPE, Storage↩
Policy >::CTYPE
```

Definition at line 32 of file proxy.h.

6.728.2.2 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::simple_name_proxy< RTYPE, Storage↩
Policy >::VECTOR
```

Definition at line 31 of file proxy.h.

6.728.3 Constructor & Destructor Documentation

6.728.3.1 simple_name_proxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::simple_name_proxy (
            VECTOR & v,
            const std::string & name_ )  [inline]
```

Definition at line 34 of file proxy.h.

6.728.3.2 simple_name_proxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::simple_name_proxy (
            const simple_name_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 36 of file proxy.h.

6.728.3.3 ∼simple_name_proxy()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::∼simple_name_proxy ( )  [inline]
```

Definition at line 38 of file proxy.h.

6.728.4 Member Function Documentation

6.728.4.1 get()

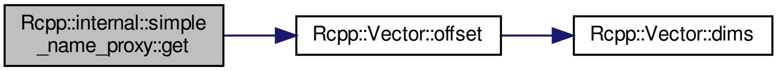
```
template<int RTYPE, template< class > class StoragePolicy>
CTYPE Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline], [private]
```

Definition at line 78 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator CTYPE(), Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator SEXP(), and Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



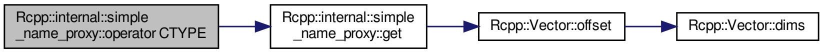
6.728.4.2 operator CTYPE()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator CTYPE ( ) const  [inline]
```

Definition at line 57 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



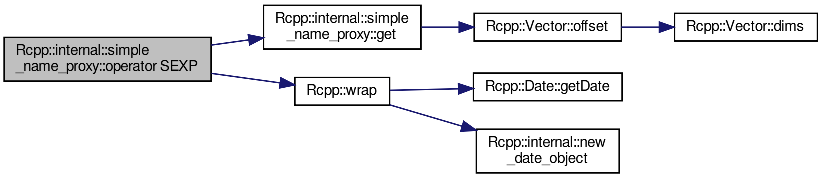
6.728.4.3 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

Definition at line 62 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::wrap().

Here is the call graph for this function:



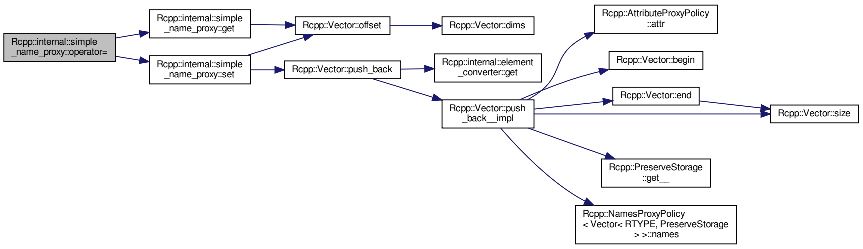
6.728.4.4 operator=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
            const simple_name_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 44 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::simple _name_proxy::operator=
Rcpp::internal::simple _name_proxy::get
Rcpp::internal::simple _name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::push_back
Rcpp::Vector::dims
Rcpp::internal::element _converter::get
Rcpp::Vector::push _back__impl
Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

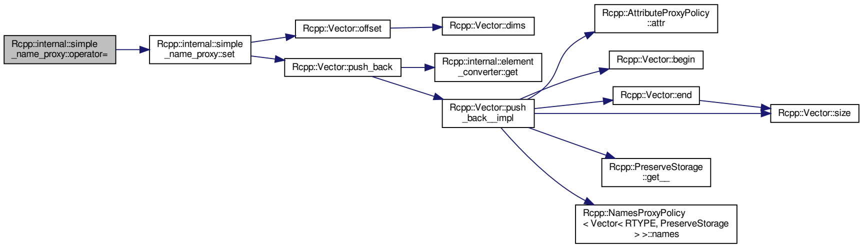
6.728.4.5 operator=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
            const T & rhs )  [inline]
```

Definition at line 50 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::simple _name_proxy::operator=
Rcpp::internal::simple _name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::push_back
Rcpp::Vector::dims
Rcpp::internal::element _converter::get
Rcpp::Vector::push _back__impl
Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

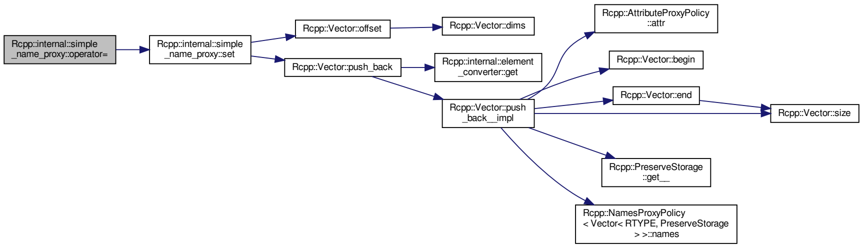
6.728.4.6 operator=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
             CTYPE rhs )  [inline]
```

Definition at line 40 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



6.728.4.7 set()

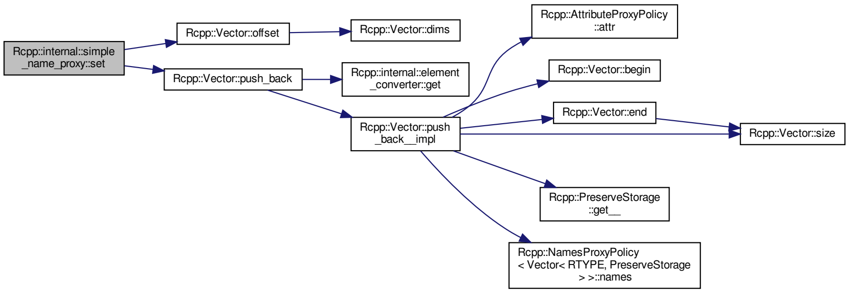
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set (
             CTYPE rhs )  [inline], [private]
```

Definition at line 69 of file proxy.h.

References Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::parent, and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Referenced by Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



6.728.5 Member Data Documentation

6.728.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::name  [private]
```

Definition at line 68 of file proxy.h.

Referenced by Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set().

6.728.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::parent  [private]
```

Definition at line 67 of file proxy.h.

Referenced by Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.729 Rcpp::SingleLogicalResult< NA, T > Class Template Reference

6.729.1 Detailed Description

template< bool NA, typename T >
class Rcpp::SingleLogicalResult< NA, T >

Definition at line 40 of file Vector.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Vector.h

6.730 Rcpp::sugar::SingleLogicalResult< NA, T > Class Template Reference

```
#include <SingleLogicalResult.h>
```

Public Member Functions

- SingleLogicalResult ()
- void apply ()
- bool is_true ()
- bool is_false ()
- bool is_na ()
- operator SEXP ()
- operator bool ()
- int size ()
- int get ()
- SEXP get_sexp ()

Static Public Attributes

- static const int UNRESOLVED = -5

Protected Member Functions

- void set (int x)
- void reset ()
- void set_true ()
- void set_false ()
- void set_na ()
- bool is_unresolved ()

Protected Attributes

- int result

6.730.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::SingleLogicalResult< NA, T >

Definition at line 42 of file SingleLogicalResult.h.

6.730.2 Constructor & Destructor Documentation

6.730.2.1 SingleLogicalResult()

```
template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::SingleLogicalResult ( ) [inline]
```

Definition at line 46 of file SingleLogicalResult.h.

6.730.3 Member Function Documentation

6.730.3.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::apply ( ) [inline]
```

Definition at line 48 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp(), Rcpp::sugar::SingleLogicalResult< NA, T >::is_false(), Rcpp::sugar::SingleLogicalResult< NA, T >::is_na(), and Rcpp::sugar::SingleLogicalResult< NA, T >::is_true().

6.730.3.2 get()

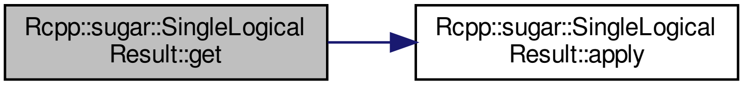
```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::get ( ) [inline]
```

Definition at line 81 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), and Rcpp::sugar::SingleLogicalResult< NA, T >::result.

Referenced by Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply(), Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply(), and Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply().

Here is the call graph for this function:



6.730.3.3 get_sexp()

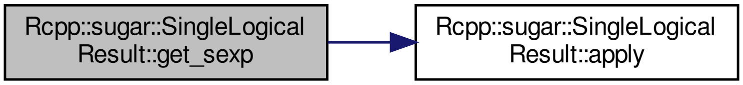
```
template<bool NA, typename T >
SEXP Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp ( )  [inline]
```

Definition at line 86 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator SEXP().

Here is the call graph for this function:



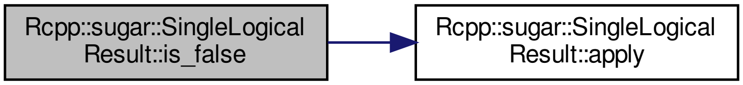
6.730.3.4 is_false()

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_false ( )  [inline]
```

Definition at line 59 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Here is the call graph for this function:



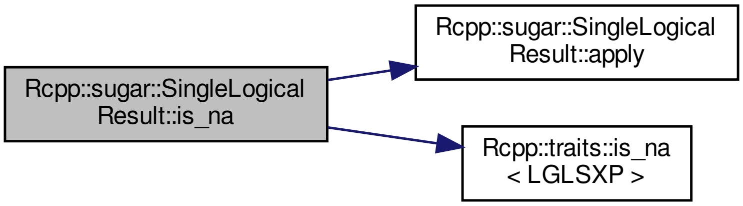
6.730.3.5 is_na()

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_na ( )  [inline]
```

Definition at line 64 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), and Rcpp::traits::is_na< LGLSXP >().

Here is the call graph for this function:



6.730.3.6 is_true()

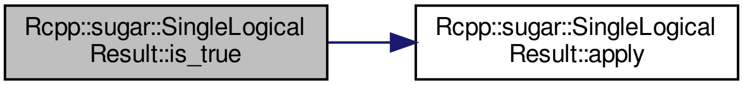
```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_true ( )  [inline]
```

Definition at line 54 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool().

Here is the call graph for this function:



6.730.3.7 is_unresolved()

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_unresolved ( )  [inline], [protected]
```

Definition at line 98 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED.

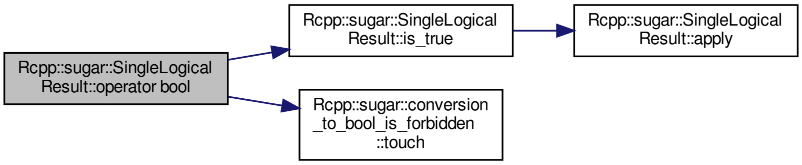
6.730.3.8 operator bool()

```
template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool ( )  [inline]
```

Definition at line 73 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::is_true(), and Rcpp::sugar::conversion_to_bool_is_forbidden< x >::touch().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::operator bool
Rcpp::sugar::SingleLogical Result::is_true
Rcpp::sugar::SingleLogical Result::apply
Rcpp::sugar::conversion _to_bool_is_forbidden ::touch

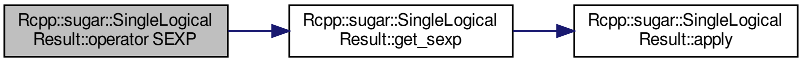
6.730.3.9 operator SEXP()

```
template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::operator SEXP ( )  [inline]
```

Definition at line 69 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::operator SEXP
Rcpp::sugar::SingleLogical Result::get_sexp
Rcpp::sugar::SingleLogical Result::apply

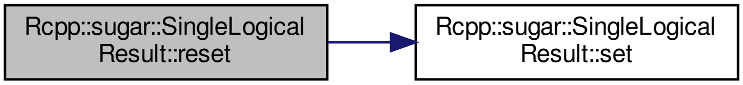
6.730.3.10 reset()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::reset ( )  [inline], [protected]
```

Definition at line 94 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set(), and Rcpp::sugar::SingleLogicalResult< NA, T >::↩ UNRESOLVED.

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::reset
Rcpp::sugar::SingleLogical Result::set

6.730.3.11 set()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set (
             int x )  [inline], [protected]
```

Definition at line 93 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply(), Rcpp::sugar::SingleLogicalResult< NA, T >::reset(), Rcpp::sugar::SingleLogicalResult< NA, T >::set_false(), Rcpp::sugar::SingleLogicalResult< NA, T >::set_na(), and Rcpp::sugar::SingleLogicalResult< NA, T >::set_true().

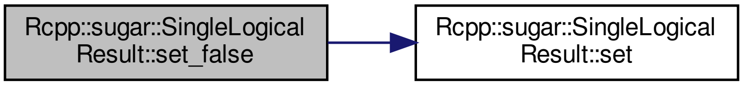
6.730.3.12 set_false()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_false ( )  [inline], [protected]
```

Definition at line 96 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::set_false
Rcpp::sugar::SingleLogical Result::set

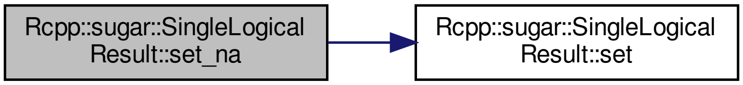
6.730.3.13 set_na()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_na ( )  [inline], [protected]
```

Definition at line 97 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::set_na
Rcpp::sugar::SingleLogical Result::set

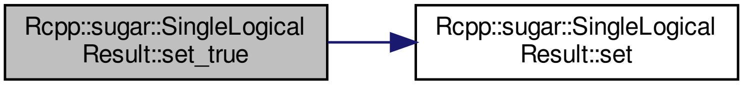
6.730.3.14 set_true()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_true ( )  [inline], [protected]
```

Definition at line 95 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set().

Here is the call graph for this function:


Rcpp::sugar::SingleLogical Result::set_true
Rcpp::sugar::SingleLogical Result::set

6.730.3.15 size()

```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::size ( )  [inline]
```

Definition at line 79 of file SingleLogicalResult.h.

6.730.4 Member Data Documentation

6.730.4.1 result

```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::result  [protected]
```

Definition at line 92 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::get().

6.730.4.2 UNRESOLVED

```
template<bool NA, typename T >
const int Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED = -5  [static]
```

Definition at line 44 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), Rcpp::sugar::SingleLogicalResult< NA, T >::is↩_unresolved(), and Rcpp::sugar::SingleLogicalResult< NA, T >::reset().

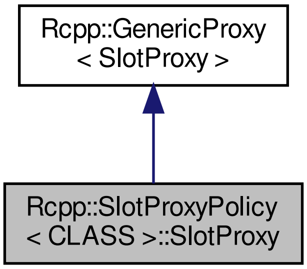
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

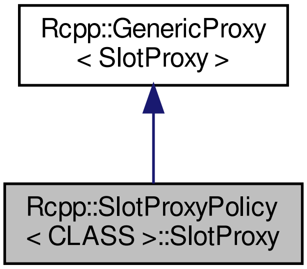
6.731 Rcpp::SlotProxyPolicy< CLASS >::SlotProxy Class Reference

```
#include <SlotProxy.h>
```

Inheritance diagram for Rcpp::SlotProxyPolicy< CLASS >::SlotProxy:



Collaboration diagram for Rcpp::SlotProxyPolicy< CLASS >::SlotProxy:



Public Member Functions

- SlotProxy (CLASS &v, const std::string &name)
- SlotProxy & operator= (const SlotProxy &rhs)
- template< typename T >
  SlotProxy & operator= (const T &rhs)
- template< typename T >
  operator T () const
- operator SEXP () const
- template< typename T >
  SlotProxyPolicy< CLASS >::SlotProxy & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- CLASS & parent
- SEXP slot_name

6.731.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >::SlotProxy

Definition at line 27 of file SlotProxy.h.

6.731.2 Constructor & Destructor Documentation

6.731.2.1 SlotProxy()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::SlotProxy (
            CLASS & v,
            const std::string & name )  [inline]
```

Definition at line 29 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot_name.

6.731.3 Member Function Documentation

6.731.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get ( ) const  [inline], [private]
```

Definition at line 51 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator SEXP(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=().

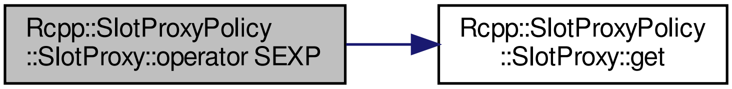
6.731.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator SEXP ( ) const  [inline]
```

Definition at line 43 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get().

Here is the call graph for this function:



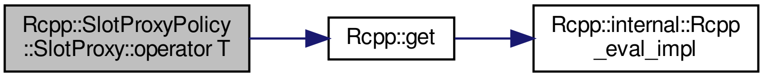
6.731.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator T
```

Definition at line 91 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



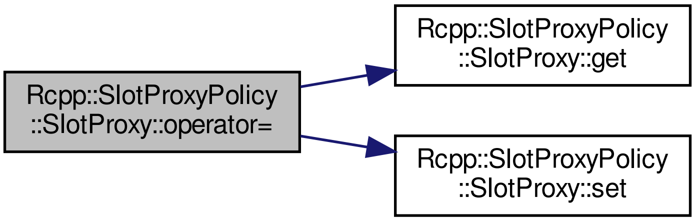
6.731.3.4 operator=() [1/3]

```
template<typename CLASS >
SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
             const SlotProxy & rhs )  [inline]
```

Definition at line 35 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::set().

Here is the call graph for this function:


Rcpp::SlotProxyPolicy ::SlotProxy::operator=
Rcpp::SlotProxyPolicy ::SlotProxy::get
Rcpp::SlotProxyPolicy ::SlotProxy::set

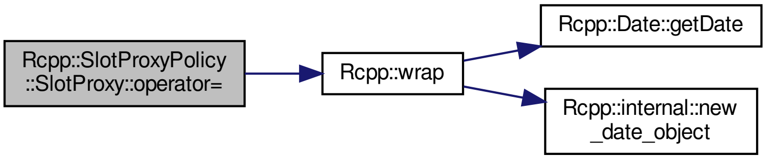
6.731.3.5 operator=() [2/3]

```
template<typename CLASS >
template<typename T >
SlotProxyPolicy<CLASS>::SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
             const T & rhs )
```

Definition at line 84 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::SlotProxyPolicy ::SlotProxy::operator=
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.731.3.6 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
             const T & rhs )
```

6.731.3.7 set()

```
template<typename CLASS >
void Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::set (
             SEXP x )  [inline], [private]
```

Definition at line 54 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↩ ::slot_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=().

6.731.4 Member Data Documentation

6.731.4.1 parent

```
template<typename CLASS >
CLASS& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent  [private]
```

Definition at line 48 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), and Rcpp::SlotProxyPolicy< CLASS >::Slot↩ Proxy::set().

6.731.4.2 slot_name

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot_name  [private]
```

Definition at line 49 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↩ ::set(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::SlotProxy().

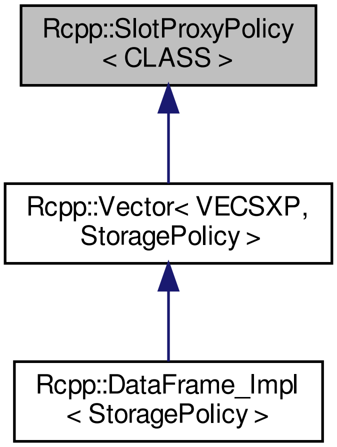
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/SlotProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.732 Rcpp::SlotProxyPolicy< CLASS > Class Template Reference

```
#include <SlotProxy.h>
```

Inheritance diagram for Rcpp::SlotProxyPolicy< CLASS >:



Classes

- class const_SlotProxy
- class SlotProxy

Public Member Functions

- SlotProxy slot (const std::string &name)
- const_SlotProxy slot (const std::string &name) const
- bool hasSlot (const std::string &name) const

6.732.1 Detailed Description

template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >

Definition at line 24 of file SlotProxy.h.

6.732.2 Member Function Documentation

6.732.2.1 hasSlot()

```
template<typename CLASS >
bool Rcpp::SlotProxyPolicy< CLASS >::hasSlot (
            const std::string & name ) const  [inline]
```

Definition at line 94 of file SlotProxy.h.

6.732.2.2 slot() [1/2]

```
template<typename CLASS >
SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (
            const std::string & name )  [inline]
```

Definition at line 83 of file SlotProxy.h.

6.732.2.3 slot() [2/2]

```
template<typename CLASS >
const_SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (
            const std::string & name ) const  [inline]
```

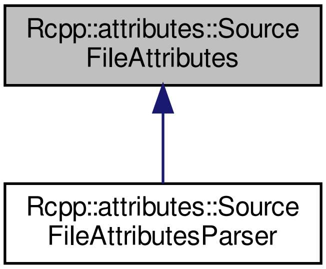
Definition at line 88 of file SlotProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/SlotProxy.h

6.733 Rcpp::attributes::SourceFileAttributes Class Reference

Inheritance diagram for Rcpp::attributes::SourceFileAttributes:


Rcpp::attributes::Source FileAttributes
Rcpp::attributes::Source FileAttributesParser

Public Types

- typedef std::vector< Attribute >::const_iterator const_iterator

Public Member Functions

- virtual ∼SourceFileAttributes ()
- virtual const std::string & sourceFile () const =0
- virtual bool hasInterface (const std::string &name) const =0
- virtual const_iterator begin () const =0
- virtual const_iterator end () const =0
- virtual const std::vector< std::string > & modules () const =0
- virtual const std::vector< std::vector< std::string > > & roxygenChunks () const =0
- virtual bool hasGeneratorOutput () const =0
- virtual bool hasPackageInit () const =0

6.733.1 Detailed Description

Definition at line 436 of file attributes.cpp.

6.733.2 Member Typedef Documentation

6.733.2.1 const_iterator

```
typedef std::vector<Attribute>::const_iterator Rcpp::attributes::SourceFileAttributes::const_iterator
```

Definition at line 443 of file attributes.cpp.

6.733.3 Constructor & Destructor Documentation

6.733.3.1 ∼SourceFileAttributes()

```
virtual Rcpp::attributes::SourceFileAttributes::∼SourceFileAttributes ( )  [inline], [virtual]
```

Definition at line 439 of file attributes.cpp.

6.733.4 Member Function Documentation

6.733.4.1 begin()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributes::begin ( ) const  [pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), and Rcpp::attributes::generateCpp().

6.733.4.2 end()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributes::end ( ) const  [pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), and Rcpp::attributes::generateCpp().

6.733.4.3 hasGeneratorOutput()

```
virtual bool Rcpp::attributes::SourceFileAttributes::hasGeneratorOutput ( ) const  [pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

6.733.4.4 hasInterface()

```
virtual bool Rcpp::attributes::SourceFileAttributes::hasInterface (
             const std::string & name ) const  [pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::RExportsGenerator::doWriteFunctions(), and Rcpp::attributes::ExportsGenerator::writeFunctions().

6.733.4.5 hasPackageInit()

```
virtual bool Rcpp::attributes::SourceFileAttributes::hasPackageInit ( ) const  [pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

6.733.4.6 modules()

```
virtual const std::vector<std::string>& Rcpp::attributes::SourceFileAttributes::modules ( ) const
[pure virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

6.733.4.7 roxygenChunks()

```
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().

6.733.4.8 sourceFile()

```
virtual const std::string& Rcpp::attributes::SourceFileAttributes::sourceFile ( ) const  [pure
virtual]
```

Implemented in Rcpp::attributes::SourceFileAttributesParser.

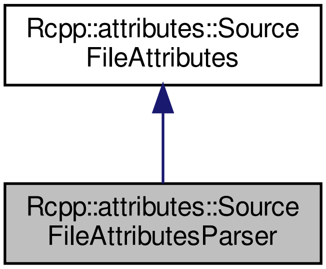
Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

The documentation for this class was generated from the following file:

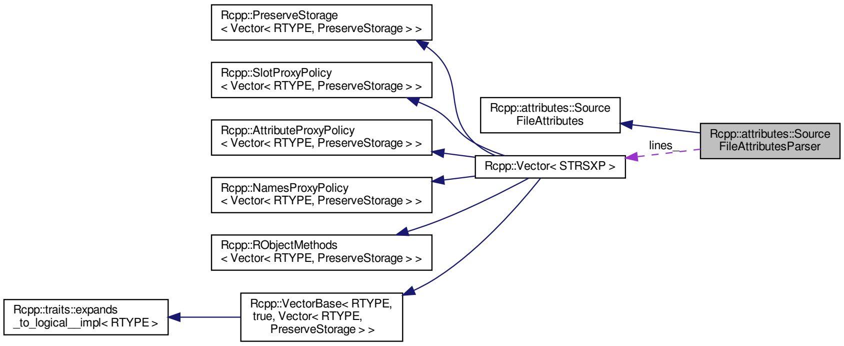
- src/attributes.cpp

6.734 Rcpp::attributes::SourceFileAttributesParser Class Reference

Inheritance diagram for Rcpp::attributes::SourceFileAttributesParser:



Collaboration diagram for Rcpp::attributes::SourceFileAttributesParser:



Public Member Functions

- SourceFileAttributesParser (const std::string &sourceFile, const std::string &packageFile, bool 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
- bool hasPackageInit () const
- const std::vector< std::string > & embeddedR () const
- const std::vector< FileInfo > & sourceDependencies () const

Private Member Functions

- SourceFileAttributesParser (const SourceFileAttributesParser &)
- SourceFileAttributesParser & operator= (const SourceFileAttributesParser &)
- Attribute parseAttribute (const std::vector< std::string > &match, int lineNumber)
- std::vector< Param > parseParameters (const std::string &input)
- Function parseFunction (size_t lineNumber)
- std::string parseSignature (size_t lineNumber)
- std::vector< std::string > parseArguments (const std::string &argText)
- Type parseType (const std::string &text)
- bool isKnownAttribute (const std::string &name) const
- void attributeWarning (const std::string &message, const std::string &attribute, size_t lineNumber)
- void attributeWarning (const std::string &message, size_t lineNumber)
- void rcppExportWarning (const std::string &message, size_t lineNumber)
- void rcppExportNoFunctionFoundWarning (size_t lineNumber)
- void rcppExportInvalidParameterWarning (const std::string &param, size_t lineNumber)
- void rcppInterfacesWarning (const std::string &message, size_t lineNumber)

Private Attributes

- std::string sourceFile_
- CharacterVector lines_
- std::vector< Attribute > attributes_
- std::vector< std::string > modules_
- bool hasPackageInit_
- std::vector< std::string > embeddedR_
- std::vector< FileInfo > sourceDependencies_
- std::vector< std::vector< std::string > > roxygenChunks_
- std::vector< std::string > roxygenBuffer_

Additional Inherited Members

6.734.1 Detailed Description

Definition at line 486 of file attributes.cpp.

6.734.2 Constructor & Destructor Documentation

6.734.2.1 SourceFileAttributesParser() [1/2]

```
Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (
            const std::string & sourceFile,
            const std::string & packageFile,
            bool parseDependencies )  [explicit]
```

6.734.2.2 SourceFileAttributesParser() [2/2]

```
Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (
            const SourceFileAttributesParser & ) [private]
```

6.734.3 Member Function Documentation

6.734.3.1 attributeWarning() [1/2]

```
void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (
            const std::string & message,
            const std::string & attribute,
            size_t lineNumber ) [private]
```

6.734.3.2 attributeWarning() [2/2]

```
void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (
            const std::string & message,
            size_t lineNumber ) [private]
```

6.734.3.3 begin()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributesParser::begin ( ) const [inline],
[virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 502 of file attributes.cpp.

References attributes_.

Referenced by compileAttributes(), and hasInterface().

6.734.3.4 embeddedR()

```
const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::embeddedR ( ) const
[inline]
```

Definition at line 542 of file attributes.cpp.

References embeddedR_.

6.734.3.5 end()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributesParser::end ( ) const  [inline],
[virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 503 of file attributes.cpp.

References attributes_.

Referenced by compileAttributes(), and hasInterface().

6.734.3.6 hasGeneratorOutput()

```
virtual bool Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput ( ) const  [inline],
[virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 514 of file attributes.cpp.

References attributes_, modules_, and roxygenChunks_.

Referenced by compileAttributes().

6.734.3.7 hasInterface()

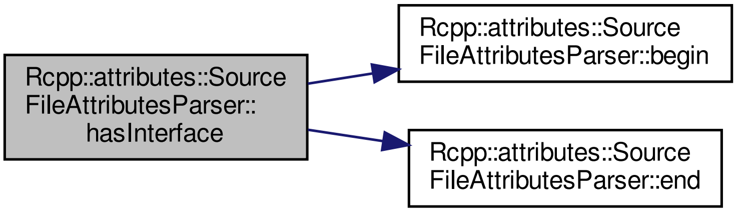
```
virtual bool Rcpp::attributes::SourceFileAttributesParser::hasInterface (
             const std::string & name ) const  [inline], [virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 521 of file attributes.cpp.

References begin(), end(), Rcpp::attributes::kInterfaceR, and Rcpp::attributes::kInterfacesAttribute.

Here is the call graph for this function:


Rcpp::attributes::Source FileAttributesParser:: hasInterface
Rcpp::attributes::Source FileAttributesParser::begin
Rcpp::attributes::Source FileAttributesParser::end

6.734.3.8 hasPackageInit()

```
bool Rcpp::attributes::SourceFileAttributesParser::hasPackageInit ( ) const  [inline], [virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 537 of file attributes.cpp.

References hasPackageInit_.

Referenced by compileAttributes().

6.734.3.9 isKnownAttribute()

```
bool Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute (
             const std::string & name ) const  [private]
```

6.734.3.10 modules()

```
virtual const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::modules ( )
const  [inline], [virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 505 of file attributes.cpp.

References modules_.

6.734.3.11 operator=()

```
SourceFileAttributesParser& Rcpp::attributes::SourceFileAttributesParser::operator= (
             const SourceFileAttributesParser &  )  [private]
```

6.734.3.12 parseArguments()

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::parseArguments (
             const std::string & argText )  [private]
```

6.734.3.13 parseAttribute()

```
Attribute Rcpp::attributes::SourceFileAttributesParser::parseAttribute (
             const std::vector< std::string > & match,
             int lineNumber )  [private]
```

6.734.3.14 parseFunction()

```
Function Rcpp::attributes::SourceFileAttributesParser::parseFunction (
             size_t lineNumber )  [private]
```

6.734.3.15 parseParameters()

```
std::vector<Param> Rcpp::attributes::SourceFileAttributesParser::parseParameters (
            const std::string & input )  [private]
```

6.734.3.16 parseSignature()

```
std::string Rcpp::attributes::SourceFileAttributesParser::parseSignature (
            size_t lineNumber )  [private]
```

6.734.3.17 parseType()

```
Type Rcpp::attributes::SourceFileAttributesParser::parseType (
            const std::string & text )  [private]
```

6.734.3.18 rcppExportInvalidParameterWarning()

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportInvalidParameterWarning (
            const std::string & param,
            size_t lineNumber )  [private]
```

6.734.3.19 rcppExportNoFunctionFoundWarning()

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportNoFunctionFoundWarning (
            size_t lineNumber )  [private]
```

6.734.3.20 rcppExportWarning()

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportWarning (
            const std::string & message,
            size_t lineNumber )  [private]
```

6.734.3.21 rcppInterfacesWarning()

```
void Rcpp::attributes::SourceFileAttributesParser::rcppInterfacesWarning (
             const std::string & message,
             size_t lineNumber )  [private]
```

6.734.3.22 roxygenChunks()

```
virtual const std::vector<std::vector<std::string> >& Rcpp::attributes::SourceFileAttributes↩
Parser::roxygenChunks ( ) const  [inline], [virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 510 of file attributes.cpp.

References roxygenChunks_.

6.734.3.23 sourceDependencies()

```
const std::vector<FileInfo>& Rcpp::attributes::SourceFileAttributesParser::sourceDependencies ( )
const  [inline]
```

Definition at line 547 of file attributes.cpp.

References sourceDependencies_.

6.734.3.24 sourceFile()

```
virtual const std::string& Rcpp::attributes::SourceFileAttributesParser::sourceFile ( ) const
[inline], [virtual]
```

Implements Rcpp::attributes::SourceFileAttributes.

Definition at line 499 of file attributes.cpp.

References sourceFile_.

6.734.4 Member Data Documentation

6.734.4.1 attributes_

```
std::vector<Attribute> Rcpp::attributes::SourceFileAttributesParser::attributes_  [private]
```

Definition at line 578 of file attributes.cpp.

Referenced by begin(), end(), and hasGeneratorOutput().

6.734.4.2 embeddedR_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::embeddedR_  [private]
```

Definition at line 581 of file attributes.cpp.

Referenced by embeddedR().

6.734.4.3 hasPackageInit_

```
bool Rcpp::attributes::SourceFileAttributesParser::hasPackageInit_  [private]
```

Definition at line 580 of file attributes.cpp.

Referenced by hasPackageInit().

6.734.4.4 lines_

```
CharacterVector Rcpp::attributes::SourceFileAttributesParser::lines_  [private]
```

Definition at line 577 of file attributes.cpp.

6.734.4.5 modules_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::modules_  [private]
```

Definition at line 579 of file attributes.cpp.

Referenced by hasGeneratorOutput(), and modules().

6.734.4.6 roxygenBuffer_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::roxygenBuffer_  [private]
```

Definition at line 584 of file attributes.cpp.

6.734.4.7 roxygenChunks_

```
std::vector<std::vector<std::string> > Rcpp::attributes::SourceFileAttributesParser::roxygen↩
Chunks_  [private]
```

Definition at line 583 of file attributes.cpp.

Referenced by hasGeneratorOutput(), and roxygenChunks().

6.734.4.8 sourceDependencies_

```
std::vector<FileInfo> Rcpp::attributes::SourceFileAttributesParser::sourceDependencies_  [private]
```

Definition at line 582 of file attributes.cpp.

Referenced by sourceDependencies().

6.734.4.9 sourceFile_

```
std::string Rcpp::attributes::SourceFileAttributesParser::sourceFile_  [private]
```

Definition at line 576 of file attributes.cpp.

Referenced by sourceFile().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.735 Rcpp::algorithm::helpers::sqrt Struct Reference

```
#include <algorithm.h>
```

Public Member Functions

- template< typename T >
  double operator() (T val)

6.735.1 Detailed Description

Definition at line 217 of file algorithm.h.

6.735.2 Member Function Documentation

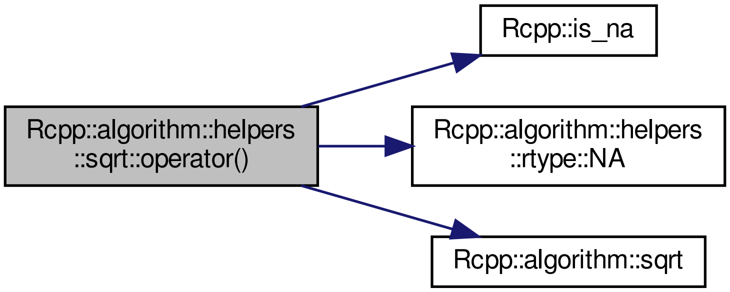
6.735.2.1 operator()()

```
template<typename T >
double Rcpp::algorithm::helpers::sqrt::operator() (
             T val )  [inline]
```

Definition at line 219 of file algorithm.h.

References Rcpp::is_na(), Rcpp::algorithm::helpers::rtype< T >::NA(), and Rcpp::algorithm::sqrt().

Here is the call graph for this function:

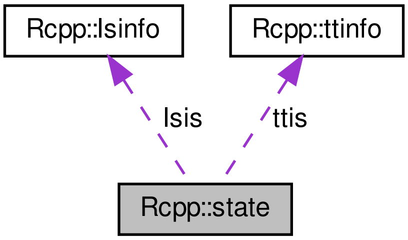

Rcpp::is_na
Rcpp::algorithm::helpers ::sqrt::operator()
Rcpp::algorithm::helpers ::rtype::NA
Rcpp::algorithm::sqrt

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/algorithm.h

6.736 Rcpp::state Struct Reference

Collaboration diagram for Rcpp::state:



Public Attributes

- int leapcnt
- int timecnt
- int typecnt
- int charcnt
- int goback
- int goahead
- time_t ats [TZ_MAX_TIMES]
- unsigned char types [TZ_MAX_TIMES]
- struct ttinfo ttis [TZ_MAX_TYPES]
- char chars [BIGGEST(BIGGEST(TZ_MAX_CHARS+1, sizeof gmt),(2 ∗(MY_TZNAME_MAX+1)))]
- struct lsinfo lsis [TZ_MAX_LEAPS]

6.736.1 Detailed Description

Definition at line 393 of file date.cpp.

6.736.2 Member Data Documentation

6.736.2.1 ats

```
time_t Rcpp::state::ats[TZ_MAX_TIMES]
```

Definition at line 400 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.2 charcnt

```
int Rcpp::state::charcnt
```

Definition at line 397 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.3 chars

```
char Rcpp::state::chars[BIGGEST(BIGGEST(TZ_MAX_CHARS+1, sizeof gmt),(2 *(MY_TZNAME_MAX+1)))]
```

Definition at line 403 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.4 goahead

```
int Rcpp::state::goahead
```

Definition at line 399 of file date.cpp.

Referenced by Rcpp::tzload().

6.736.2.5 goback

```
int Rcpp::state::goback
```

Definition at line 398 of file date.cpp.

Referenced by Rcpp::tzload().

6.736.2.6 leapcnt

```
int Rcpp::state::leapcnt
```

Definition at line 394 of file date.cpp.

Referenced by Rcpp::timesub(), Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.7 lsis

```
struct lsinfo Rcpp::state::lsis[TZ_MAX_LEAPS]
```

Definition at line 403 of file date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

6.736.2.8 timecnt

```
int Rcpp::state::timecnt
```

Definition at line 395 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.9 ttis

```
struct ttinfo Rcpp::state::ttis[TZ_MAX_TYPES]
```

Definition at line 401 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.10 typecnt

```
int Rcpp::state::typecnt
```

Definition at line 396 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.736.2.11 types

```
unsigned char Rcpp::state::types[TZ_MAX_TIMES]
```

Definition at line 401 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

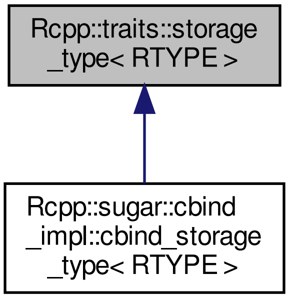
The documentation for this struct was generated from the following file:

- src/date.cpp

6.737 Rcpp::traits::storage_type< RTYPE > Struct Template Reference

```
#include <storage_type.h>
```

Inheritance diagram for Rcpp::traits::storage_type< RTYPE >:



Public Types

- typedef SEXP type

6.737.1 Detailed Description

template<int RTYPE>
struct Rcpp::traits::storage_type< RTYPE >

Indicates the storage type associated with a SEXP type for example: storage_type<INTSXP>::type is a typedef to int

The default is SEXP, which works for VECSXP, EXPRSXP and STRSXP

Definition at line 35 of file storage_type.h.

6.737.2 Member Typedef Documentation

6.737.2.1 type

```
template<int RTYPE>
typedef SEXP Rcpp::traits::storage_type< RTYPE >::type
```

Definition at line 36 of file storage_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage_type.h

6.738 Rcpp::traits::storage_type< CPLXSXP > Struct Reference

```
#include <storage_type.h>
```

Public Types

- typedef Rcomplex type

6.738.1 Detailed Description

Total specialization for numeric vectors (CPLXSXP) typedef to Rcomplex

Definition at line 59 of file storage_type.h.

6.738.2 Member Typedef Documentation

6.738.2.1 type

```
typedef Rcomplex Rcpp::traits::storage_type< CPLXSXP >::type
```

Definition at line 60 of file storage_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage_type.h

6.739 Rcpp::traits::storage_type< INTSXP > Struct Reference

```
#include <storage_type.h>
```

Public Types

- typedef int type

6.739.1 Detailed Description

Total specialization for integer vector (INTSXP) typedef to int

Definition at line 43 of file storage_type.h.

6.739.2 Member Typedef Documentation

6.739.2.1 type

```
typedef int Rcpp::traits::storage_type< INTSXP >::type
```

Definition at line 44 of file storage_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage_type.h

6.740 Rcpp::traits::storage_type< LGLSXP > Struct Reference

```
#include <storage_type.h>
```

Public Types

- typedef int type

6.740.1 Detailed Description

Total specialization for logical vectors (LGLSXP) typedef to int

Definition at line 75 of file storage_type.h.

6.740.2 Member Typedef Documentation

6.740.2.1 type

```
typedef int Rcpp::traits::storage_type< 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

6.741 Rcpp::traits::storage_type< RAWSXP > Struct Reference

```
#include <storage_type.h>
```

Public Types

- typedef Rbyte type

6.741.1 Detailed Description

Total specialization for raw vectors (RAWSXP) typedef to Rbyte

Definition at line 67 of file storage_type.h.

6.741.2 Member Typedef Documentation

6.741.2.1 type

```
typedef Rbyte Rcpp::traits::storage_type< 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

6.742 Rcpp::traits::storage_type< REALSXP > Struct Reference

```
#include <storage_type.h>
```

Public Types

- typedef double type

6.742.1 Detailed Description

Total specialization for numeric vectors (REALSXP) typedef to double

Definition at line 51 of file storage_type.h.

6.742.2 Member Typedef Documentation

6.742.2.1 type

```
typedef double Rcpp::traits::storage_type< REALSXP >::type
```

Definition at line 52 of file storage_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage_type.h

6.743 Rcpp::String Class Reference

```
#include <String.h>
```

Public Types

- typedef internal::string_proxy< STRSXP > StringProxy
- typedef internal::const_string_proxy< STRSXP > const_StringProxy

Public Member Functions

- String ()
- String (const String &s)
- String (SEXP 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_impl () const
- SEXP get_sexp () const
- SEXP get_sexp ()
- operator std::string () const
- operator std::wstring () const
- const char ∗ get_cstring () const
- cetype_t get_encoding () const
- void set_encoding (cetype_t encoding)
- bool operator< (const Rcpp::String &other) const
- bool operator== (const Rcpp::String &other) const
- bool operator!= (const Rcpp::String &other) const
- bool operator== (const StringProxy &other) const
- bool operator!= (const StringProxy &other) const
- bool operator== (const const_StringProxy &other) const
- bool operator!= (const const_StringProxy &other) const
- bool operator> (const Rcpp::String &other) const
- bool operator== (SEXP other) const
- bool operator!= (SEXP other) const

Private Member Functions

- template<typename T >
  String & assign_wide_string (const T &s)
- template<typename T >
  String & append_wide_string (const T &s)
- bool is_na () const
- void setBuffer ()
- void setData ()
- template<typename T >
  void append (const T &s)

Private Attributes

- SEXP data
- SEXP token
- std::string buffer
- bool valid
- bool buffer_ready
- cetype_t enc

6.743.1 Detailed Description

A single string, i.e. an element of a character vector. This represents CHARSXP SEXP

Definition at line 49 of file String.h.

6.743.2 Member Typedef Documentation

6.743.2.1 const_StringProxy

```
typedef internal::const_string_proxy<STRSXP> Rcpp::String::const_StringProxy
```

Definition at line 52 of file String.h.

6.743.2.2 StringProxy

```
typedef internal::string_proxy<STRSXP> Rcpp::String::StringProxy
```

Definition at line 51 of file String.h.

6.743.3 Constructor & Destructor Documentation

6.743.3.1 String() [1/16]

```
Rcpp::String::String ( )  [inline]
```

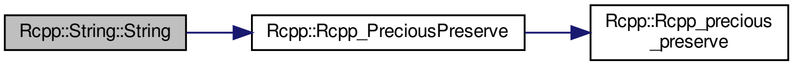
default constructor

Definition at line 55 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Referenced by replace_all().

Here is the call graph for this function:



6.743.3.2 String() [2/16]

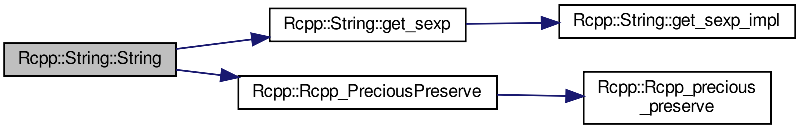
```
Rcpp::String::String (
             const String & s )  [inline]
```

copy constructor

Definition at line 61 of file String.h.

References buffer_ready, data, get_sexp(), Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Here is the call graph for this function:



6.743.3.3 String() [3/16]

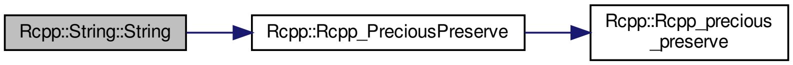
```
Rcpp::String::String (
             SEXP charsxp )  [inline]
```

construct a string from a single CHARSXP SEXP

Definition at line 70 of file String.h.

References buffer_ready, data, enc, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, token, and valid.

Here is the call graph for this function:



6.743.3.4 String() [4/16]

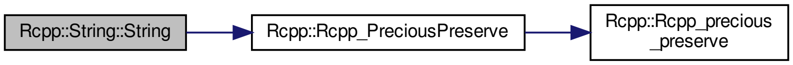
```
Rcpp::String::String (
             const StringProxy & proxy )  [inline]
```

from string proxy

Definition at line 93 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Here is the call graph for this function:



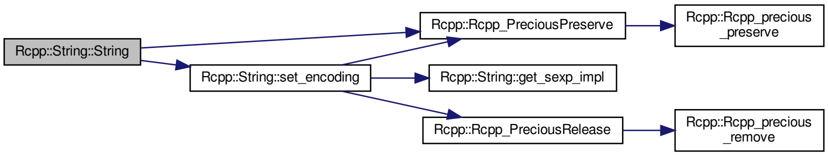
6.743.3.5 String() [5/16]

```
Rcpp::String::String (
             const StringProxy & proxy,
             cetype_t enc )  [inline]
```

Definition at line 98 of file String.h.

References data, enc, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, set_encoding(), and token.

Here is the call graph for this function:


Rcpp::String::String
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious
_preserve
Rcpp::String::set_encoding
Rcpp::String::get_sexp_impl
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious
_remove

6.743.3.6 String() [6/16]

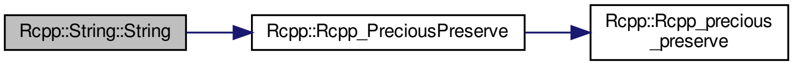
```
Rcpp::String::String (
            const const_StringProxy & proxy )  [inline]
```

from string proxy

Definition at line 105 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Here is the call graph for this function:



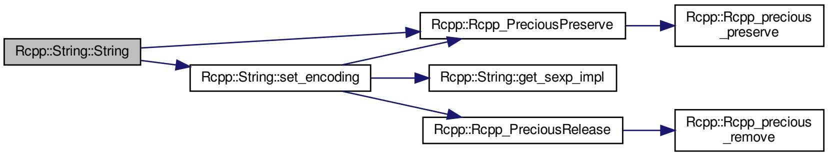
6.743.3.7 String() [7/16]

```
Rcpp::String::String (
            const const_StringProxy & proxy,
            cetype_t enc )  [inline]
```

Definition at line 110 of file String.h.

References data, enc, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, set_encoding(), and token.

Here is the call graph for this function:


Rcpp::String::String
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious
_preserve
Rcpp::String::set_encoding
Rcpp::String::get_sexp_impl
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious
_remove

6.743.3.8 String() [8/16]

```
Rcpp::String::String (
            const std::string & s,
            cetype_t enc = CE_UTF8 )  [inline]
```

from a std::string

Definition at line 117 of file String.h.

References RCPP_STRING_DEBUG.

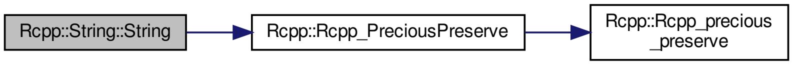
6.743.3.9 String() [9/16]

```
Rcpp::String::String (
            const std::wstring & s,
            cetype_t enc = CE_UTF8 )  [inline]
```

Definition at line 138 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Here is the call graph for this function:



6.743.3.10 String() [10/16]

```
Rcpp::String::String (
            const char * s,
            cetype_t enc = CE_UTF8 )  [inline]
```

from a const char∗

Definition at line 144 of file String.h.

References data, RCPP_STRING_DEBUG, and token.

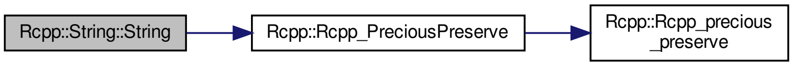
6.743.3.11 String() [11/16]

```
Rcpp::String::String (
            const wchar_t * s,
            cetype_t enc = CE_UTF8 )  [inline]
```

Definition at line 150 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, and token.

Here is the call graph for this function:



6.743.3.12 String() [12/16]

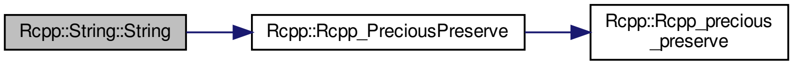
```
Rcpp::String::String (
            int x )  [inline]
```

constructors from R primitives

Definition at line 156 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), and token.

Here is the call graph for this function:



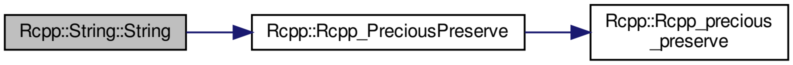
6.743.3.13 String() [13/16]

```
Rcpp::String::String (
             double x )  [inline]
```

Definition at line 159 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), and token.

Here is the call graph for this function:



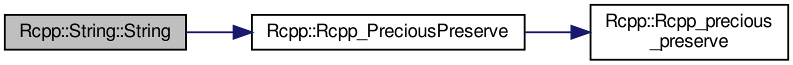
6.743.3.14 String() [14/16]

```
Rcpp::String::String (
             bool x )  [inline]
```

Definition at line 162 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), and token.

Here is the call graph for this function:



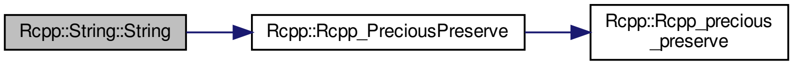
6.743.3.15 String() [15/16]

```
Rcpp::String::String (
             Rcomplex x )  [inline]
```

Definition at line 165 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), and token.

Here is the call graph for this function:



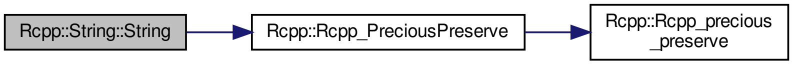
6.743.3.16 String() [16/16]

```
Rcpp::String::String (
             Rbyte x )  [inline]
```

Definition at line 168 of file String.h.

References data, Rcpp::Rcpp_PreciousPreserve(), and token.

Here is the call graph for this function:



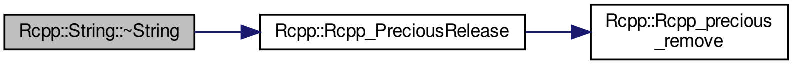
6.743.3.17 ∼String()

```
Rcpp::String::∼String ( )  [inline]
```

Definition at line 172 of file String.h.

References data, Rcpp::Rcpp_PreciousRelease(), and token.

Here is the call graph for this function:



6.743.4 Member Function Documentation

6.743.4.1 append()

```
template<typename T >
void Rcpp::String::append (
             const T & s )  [inline], [private]
```

Definition at line 672 of file String.h.

References buffer.

6.743.4.2 append_wide_string()

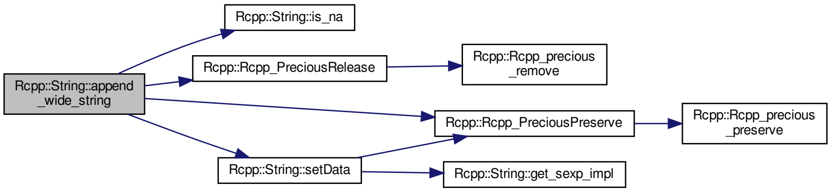
```
template<typename T >
String& Rcpp::String::append_wide_string (
             const T & s )  [inline], [private]
```

Definition at line 331 of file String.h.

References buffer_ready, data, DEMANGLE, is_na(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_Precious←Release(), RCPP_STRING_DEBUG_1, setData(), token, and valid.

Referenced by operator+=().

Here is the call graph for this function:



6.743.4.3 assign_wide_string()

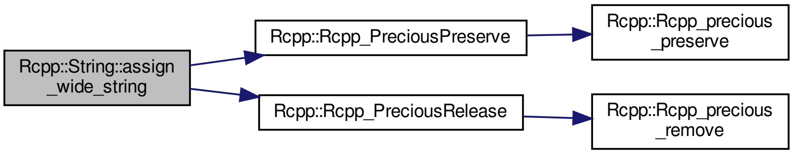
```
template<typename T >
String& Rcpp::String::assign_wide_string (
             const T & s )  [inline], [private]
```

Definition at line 303 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Referenced by operator=().

Here is the call graph for this function:



6.743.4.4 get_cstring()

```
const char* Rcpp::String::get_cstring ( ) const  [inline]
```

Definition at line 576 of file String.h.

References buffer, buffer_ready, and data.

Referenced by operator std::string(), operator std::wstring(), operator<(), operator>(), push_back(), push_front(), replace_all(), replace_first(), replace_last(), and Rcpp::trimws().

6.743.4.5 get_encoding()

```
cetype_t Rcpp::String::get_encoding ( ) const  [inline]
```

Definition at line 580 of file String.h.

References enc.

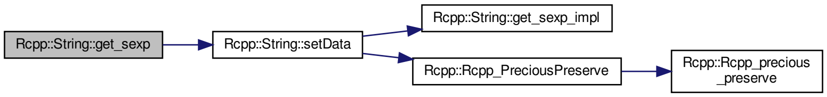
6.743.4.6 get_sexp() [1/2]

```
SEXP Rcpp::String::get_sexp ( )  [inline]
```

Definition at line 562 of file String.h.

References data, RCPP_STRING_DEBUG_1, setData(), and valid.

Here is the call graph for this function:


Rcpp::String::get_sexp
Rcpp::String::setData
Rcpp::String::get_sexp_impl
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve

6.743.4.7 get_sexp() [2/2]

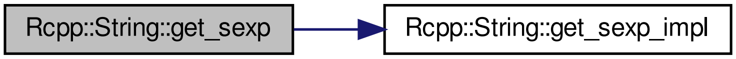
```
SEXP Rcpp::String::get_sexp ( ) const  [inline]
```

Definition at line 557 of file String.h.

References data, get_sexp_impl(), RCPP_STRING_DEBUG_1, and valid.

Referenced by Rcpp::internal::string_element_converter< RTYPE >::get(), Rcpp::internal::make_charsexp< Rcpp::↩ String >(), operator!=(), operator=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator=(), operator==(), Rcpp::internal::range_wrap_dispatch___impl__pair(), String(), and Rcpp::trimws().

Here is the call graph for this function:



6.743.4.8 get_sexp_impl()

```
SEXP Rcpp::String::get_sexp_impl ( ) const  [inline]
```

Definition at line 544 of file String.h.

References buffer, and enc.

Referenced by get_sexp(), set_encoding(), and setData().

6.743.4.9 is_na()

```
bool Rcpp::String::is_na ( ) const  [inline], [private]
```

Definition at line 657 of file String.h.

References data.

Referenced by append_wide_string(), operator+=(), push_back(), push_front(), replace_all(), replace_first(), and replace_last().

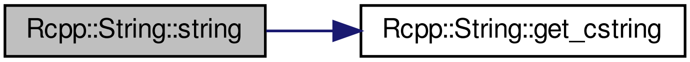
6.743.4.10 operator std::string()

```
Rcpp::String::operator std::string ( ) const  [inline]
```

Definition at line 567 of file String.h.

References get_cstring().

Here is the call graph for this function:



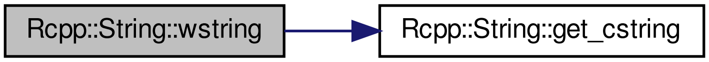
6.743.4.11 operator std::wstring()

```
Rcpp::String::operator std::wstring ( ) const  [inline]
```

Definition at line 571 of file String.h.

References get_cstring().

Here is the call graph for this function:



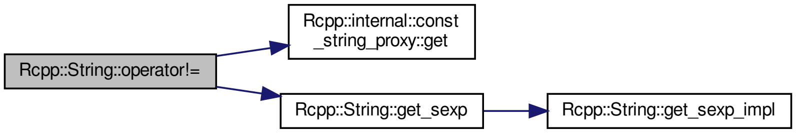
6.743.4.12 operator"!=() [1/4]

```
bool Rcpp::String::operator!= (
              const const_StringProxy & other ) const  [inline]
```

Definition at line 623 of file String.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get(), and get_sexp().

Here is the call graph for this function:



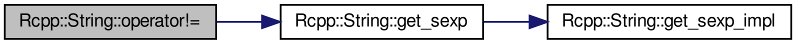
6.743.4.13 operator"!=() [2/4]

```
bool Rcpp::String::operator!= (
            const Rcpp::String & other ) const  [inline]
```

Definition at line 607 of file String.h.

References get_sexp().

Here is the call graph for this function:



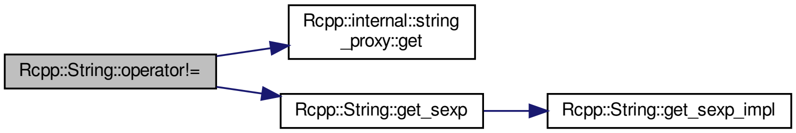
6.743.4.14 operator"!=() [3/4]

```
bool Rcpp::String::operator!= (
            const StringProxy & other ) const  [inline]
```

Definition at line 615 of file String.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), and get_sexp().

Here is the call graph for this function:



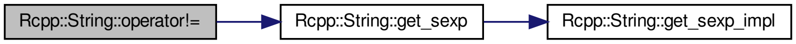
6.743.4.15 operator"!=() [4/4]

```
bool Rcpp::String::operator!= (
             SEXP other ) const  [inline]
```

Definition at line 635 of file String.h.

References get_sexp().

Here is the call graph for this function:



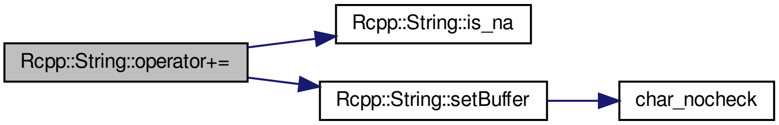
6.743.4.16 operator+=() [1/8]

```
String& Rcpp::String::operator+= (
             const char * s )  [inline]
```

Definition at line 323 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:



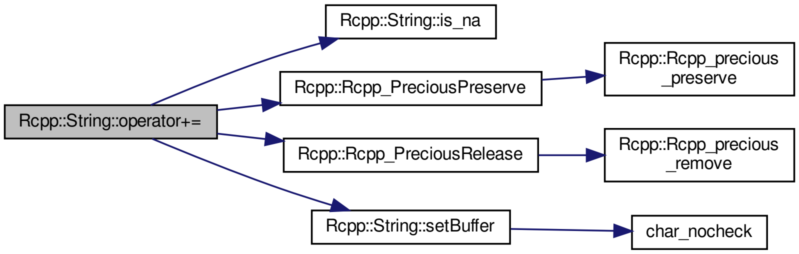
6.743.4.17 operator+=() [2/8]

```
String& Rcpp::String::operator+= (
             const const_StringProxy & proxy )  [inline]
```

Definition at line 380 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), RCPP_STRING_DEBUG, setBuffer(), token, and valid.

Here is the call graph for this function:


Rcpp::String::is_na
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_PreciousPreserve
Rcpp::String::operator+=
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove
Rcpp::String::setBuffer
char_nocheck

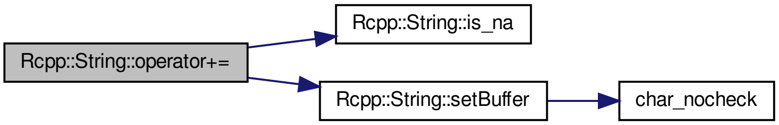
6.743.4.18 operator+=() [3/8]

```
String& Rcpp::String::operator+= (
             const std::string & s )  [inline]
```

Definition at line 316 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG, setBuffer(), and valid.

Here is the call graph for this function:



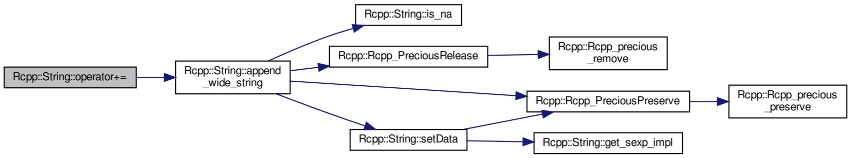
6.743.4.19 operator+=() [4/8]

```
String& Rcpp::String::operator+= (
            const std::wstring & s )  [inline]
```

Definition at line 348 of file String.h.

References append_wide_string().

Here is the call graph for this function:



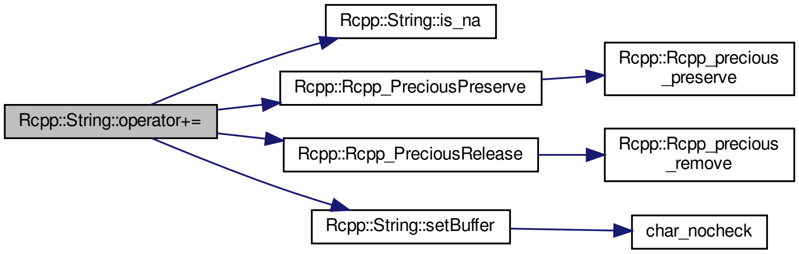
6.743.4.20 operator+=() [5/8]

```
String& Rcpp::String::operator+= (
            const String & other )  [inline]
```

Definition at line 351 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), RCPP_STRING_DEBUG, setBuffer(), token, and valid.

Here is the call graph for this function:



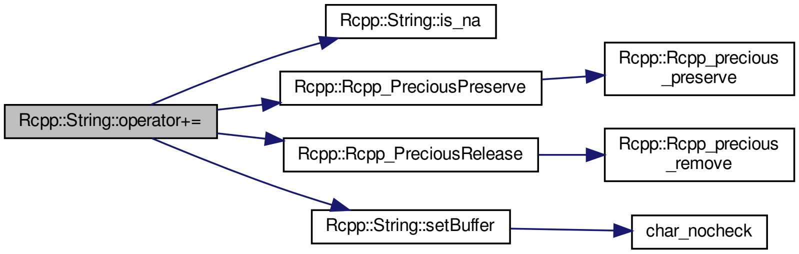
6.743.4.21 operator+=() [6/8]

```
String& Rcpp::String::operator+= (
             const StringProxy & proxy )   [inline]
```

Definition at line 365 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), RCPP_STRING_DEBUG, setBuffer(), token, and valid.

Here is the call graph for this function:


Rcpp::String::is_na
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve
Rcpp::String::operator+=
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove
Rcpp::String::setBuffer
char_nocheck

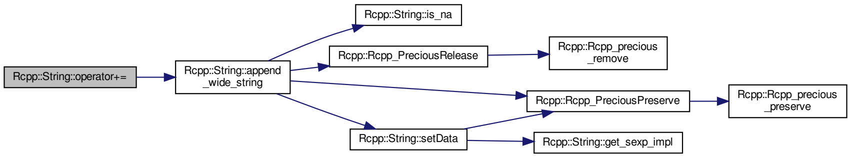
6.743.4.22 operator+=() [7/8]

```
String& Rcpp::String::operator+= (
             const wchar_t * s )   [inline]
```

Definition at line 349 of file String.h.

References append_wide_string().

Here is the call graph for this function:


Rcpp::String::is_na
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove
Rcpp::String::operator+=
Rcpp::String::append_wide_string
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve
Rcpp::String::setData
Rcpp::String::get_sexp_impl

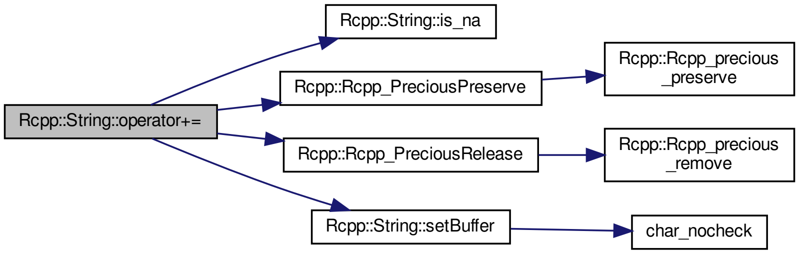
6.743.4.23 operator+=() [8/8]

```
String& Rcpp::String::operator+= (
             SEXP x )  [inline]
```

Definition at line 395 of file String.h.

References buffer, buffer_ready, data, is_na(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), RCPP_STRING_DEBUG, setBuffer(), token, and valid.

Here is the call graph for this function:



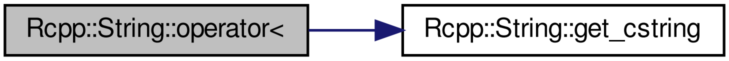
6.743.4.24 operator<()

```
bool Rcpp::String::operator< (
             const Rcpp::String & other ) const  [inline]
```

Definition at line 600 of file String.h.

References get_cstring().

Here is the call graph for this function:



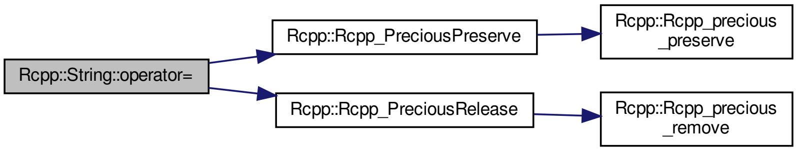
6.743.4.25 operator=() [1/12]

```
String& Rcpp::String::operator= (
             bool x )  [inline]
```

Definition at line 203 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:



6.743.4.26 operator=() [2/12]

```
String& Rcpp::String::operator= (
             const char * s )  [inline]
```

Definition at line 294 of file String.h.

References buffer, buffer_ready, and valid.

6.743.4.27 operator=() [3/12]

```
String& Rcpp::String::operator= (
             const std::string & s )  [inline]
```

Definition at line 264 of file String.h.

References buffer, buffer_ready, and valid.

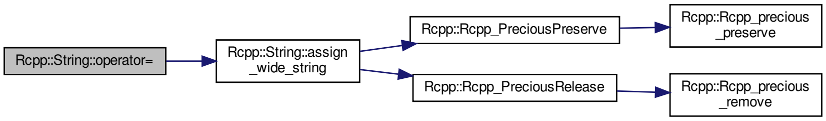
6.743.4.28 operator=() [4/12]

```
String& Rcpp::String::operator= (
             const std::wstring & s )   [inline]
```

Definition at line 313 of file String.h.

References assign_wide_string().

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::String::assign_wide_string
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_precious_remove

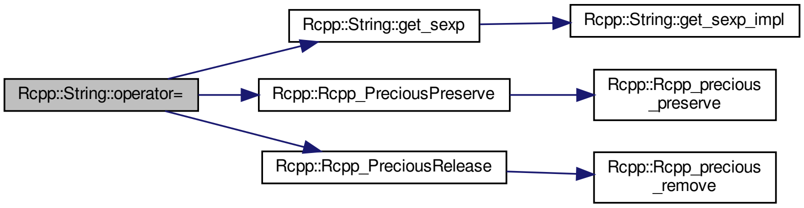
6.743.4.29 operator=() [5/12]

```
String& Rcpp::String::operator= (
             const String & other )   [inline]
```

Definition at line 240 of file String.h.

References buffer, buffer_ready, data, enc, get_sexp(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::String::get_sexp
Rcpp::String::get_sexp_impl
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove

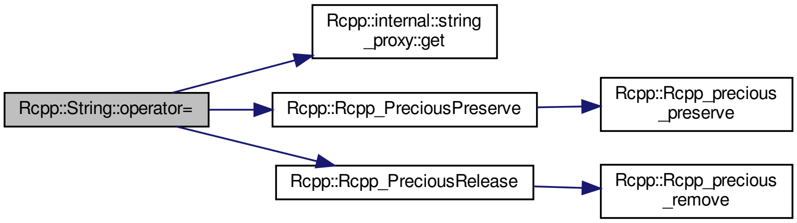
6.743.4.30 operator=() [6/12]

```
String& Rcpp::String::operator= (
             const StringProxy & proxy )  [inline]
```

Definition at line 229 of file String.h.

References buffer_ready, data, Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:



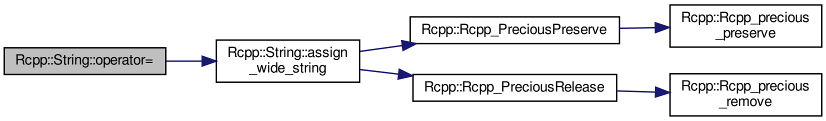
6.743.4.31 operator=() [7/12]

```
String& Rcpp::String::operator= (
             const wchar_t * s )  [inline]
```

Definition at line 314 of file String.h.

References assign_wide_string().

Here is the call graph for this function:



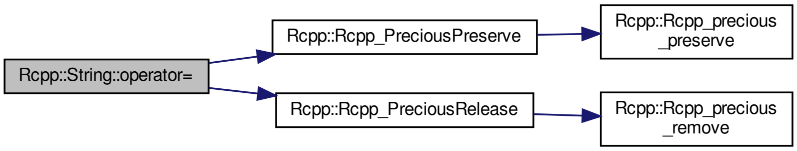
6.743.4.32 operator=() [8/12]

```
String& Rcpp::String::operator= (
            double x )  [inline]
```

Definition at line 187 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_precious_remove

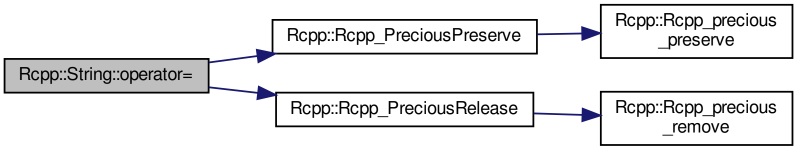
6.743.4.33 operator=() [9/12]

```
String& Rcpp::String::operator= (
            int x )  [inline]
```

Definition at line 179 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_precious_remove

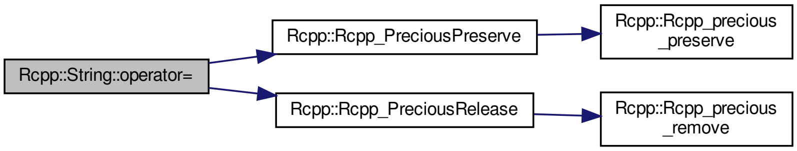
6.743.4.34 operator=() [10/12]

```
String& Rcpp::String::operator= (
             Rbyte x )  [inline]
```

Definition at line 195 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious _preserve
Rcpp::Rcpp_precious _remove

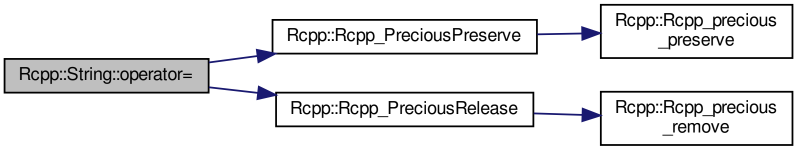
6.743.4.35 operator=() [11/12]

```
String& Rcpp::String::operator= (
             Rcomplex x )  [inline]
```

Definition at line 211 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:


Rcpp::String::operator=
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious _preserve
Rcpp::Rcpp_precious _remove

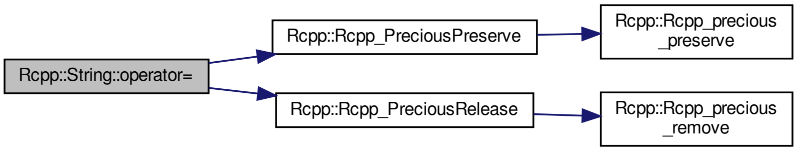
6.743.4.36 operator=() [12/12]

```
String& Rcpp::String::operator= (
             SEXP x )  [inline]
```

Definition at line 219 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Here is the call graph for this function:



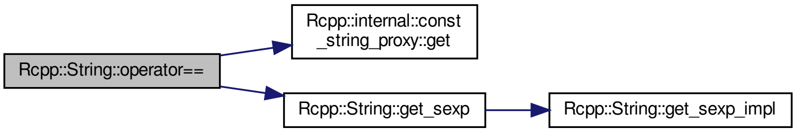
6.743.4.37 operator==() [1/4]

```
bool Rcpp::String::operator== (
             const const_StringProxy & other ) const  [inline]
```

Definition at line 619 of file String.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get(), and get_sexp().

Here is the call graph for this function:



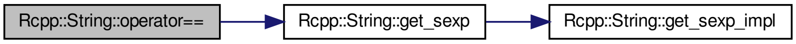
6.743.4.38 operator==() [2/4]

```
bool Rcpp::String::operator== (
            const Rcpp::String & other ) const  [inline]
```

Definition at line 604 of file String.h.

References get_sexp().

Here is the call graph for this function:



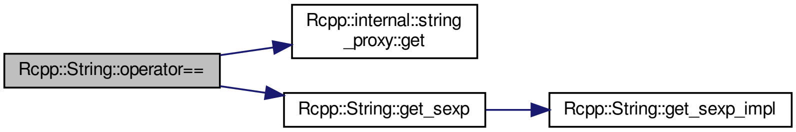
6.743.4.39 operator==() [3/4]

```
bool Rcpp::String::operator== (
            const StringProxy & other ) const  [inline]
```

Definition at line 611 of file String.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), and get_sexp().

Here is the call graph for this function:



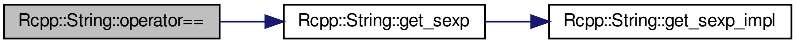
6.743.4.40 operator==() [4/4]

```
bool Rcpp::String::operator== (
             SEXP other ) const  [inline]
```

Definition at line 631 of file String.h.

References get_sexp().

Here is the call graph for this function:



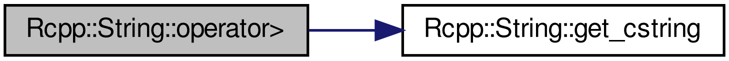
6.743.4.41 operator>()

```
bool Rcpp::String::operator> (
             const Rcpp::String & other ) const  [inline]
```

Definition at line 627 of file String.h.

References get_cstring().

Here is the call graph for this function:



6.743.4.42 push_back() [1/3]

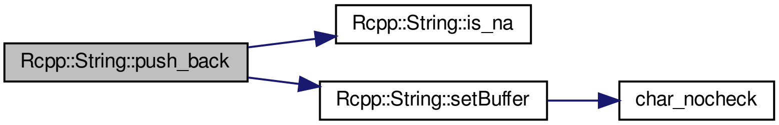
```
String& Rcpp::String::push_back (
            const char * s ) [inline]
```

Definition at line 506 of file String.h.

References buffer, is_na(), setBuffer(), and valid.

Referenced by push_back().

Here is the call graph for this function:



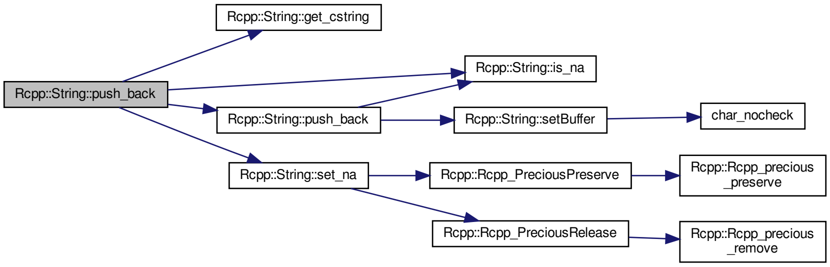
6.743.4.43 push_back() [2/3]

```
String& Rcpp::String::push_back (
            const Rcpp::String & s ) [inline]
```

Definition at line 514 of file String.h.

References get_cstring(), is_na(), push_back(), and set_na().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::is_na
Rcpp::String::push_back
Rcpp::String::push_back
Rcpp::String::setBuffer
char_nocheck
Rcpp::String::set_na
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove

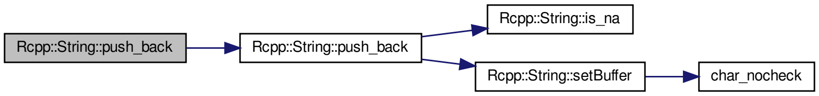
6.743.4.44 push_back() [3/3]

```
String& Rcpp::String::push_back (
             const std::string & s )  [inline]
```

Definition at line 511 of file String.h.

References push_back().

Here is the call graph for this function:



6.743.4.45 push_front() [1/3]

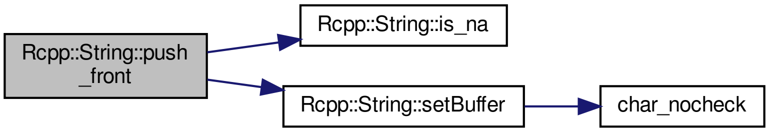
```
String& Rcpp::String::push_front (
             const char * s )  [inline]
```

Definition at line 520 of file String.h.

References buffer, is_na(), setBuffer(), and valid.

Referenced by push_front().

Here is the call graph for this function:



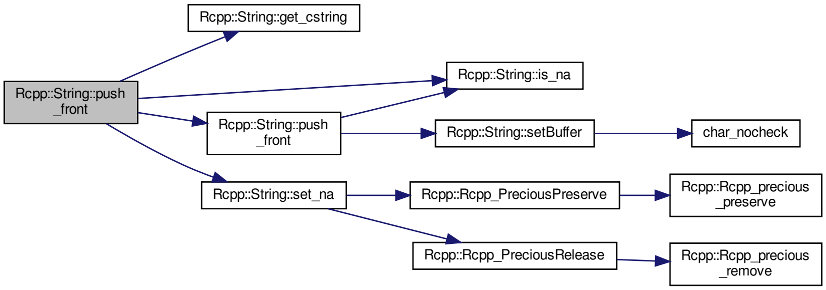
6.743.4.46 push_front() [2/3]

```
String& Rcpp::String::push_front (
            const Rcpp::String & s )  [inline]
```

Definition at line 528 of file String.h.

References get_cstring(), is_na(), push_front(), and set_na().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::push_front
Rcpp::String::is_na
Rcpp::String::push_front
Rcpp::String::setBuffer
char_nocheck
Rcpp::String::set_na
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_remove

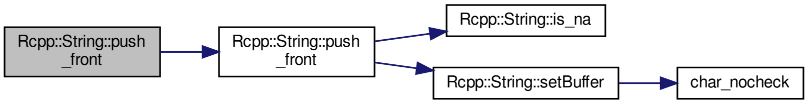
6.743.4.47 push_front() [3/3]

```
String& Rcpp::String::push_front (
            const std::string & s )  [inline]
```

Definition at line 525 of file String.h.

References push_front().

Here is the call graph for this function:



6.743.4.48 replace_all() [1/5]

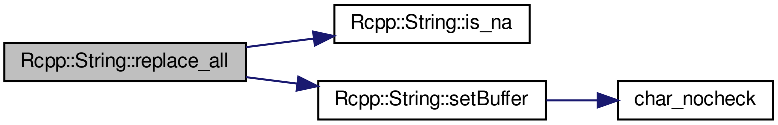
```
String& Rcpp::String::replace_all (
            const char * s,
            const char * news )  [inline]
```

Definition at line 469 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG_2, setBuffer(), and valid.

Referenced by replace_all().

Here is the call graph for this function:



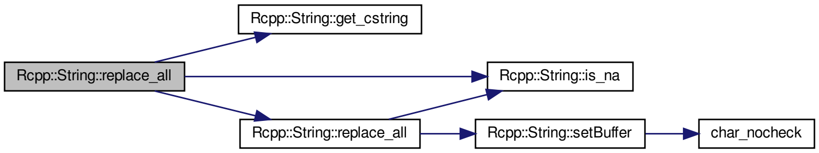
6.743.4.49 replace_all() [2/5]

```
String& Rcpp::String::replace_all (
            const char * s,
            const Rcpp::String & news )  [inline]
```

Definition at line 495 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:


Rcpp::String::replace_all
Rcpp::String::get_cstring
Rcpp::String::is_na
Rcpp::String::replace_all
Rcpp::String::setBuffer
char_nocheck

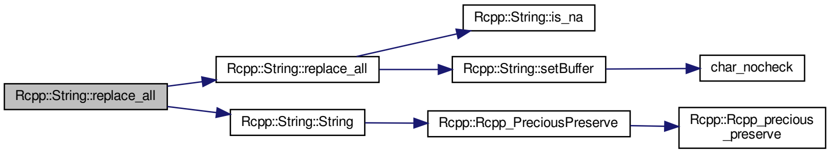
6.743.4.50 replace_all() [3/5]

```
template<typename LHS , typename RHS >
String& Rcpp::String::replace_all (
             const LHS & s,
             const RHS & news )  [inline]
```

Definition at line 486 of file String.h.

References replace_all(), and String().

Here is the call graph for this function:


Rcpp::String::replace_all
Rcpp::String::is_na
Rcpp::String::replace_all
Rcpp::String::setBuffer
char_nocheck
Rcpp::String::String
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_precious_preserve

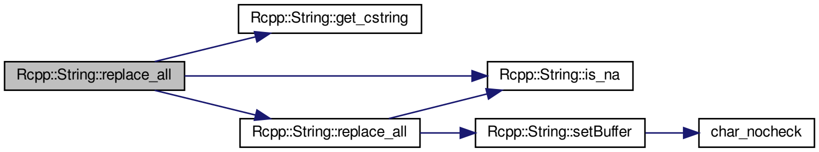
6.743.4.51 replace_all() [4/5]

```
String& Rcpp::String::replace_all (
             const Rcpp::String & s,
             const char * news )  [inline]
```

Definition at line 490 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::replace_all
Rcpp::String::is_na
Rcpp::String::replace_all
Rcpp::String::setBuffer
char_nocheck

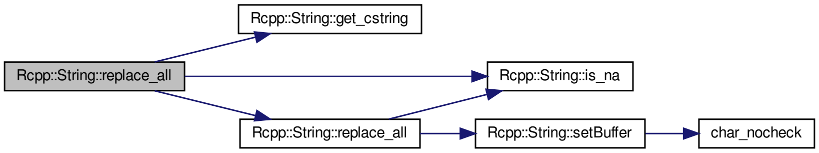
6.743.4.52 replace_all() [5/5]

```
String& Rcpp::String::replace_all (
             const Rcpp::String & s,
             const Rcpp::String & news )  [inline]
```

Definition at line 500 of file String.h.

References get_cstring(), is_na(), and replace_all().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::replace_all
Rcpp::String::is_na
Rcpp::String::replace_all
Rcpp::String::setBuffer
char_nocheck

6.743.4.53 replace_first() [1/4]

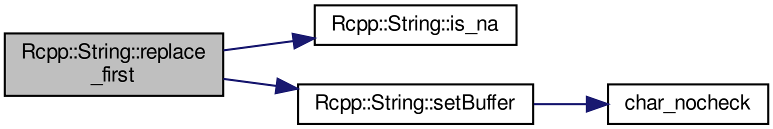
```
String& Rcpp::String::replace_first (
             const char * s,
             const char * news )  [inline]
```

Definition at line 416 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG_2, setBuffer(), and valid.

Referenced by replace_first().

Here is the call graph for this function:



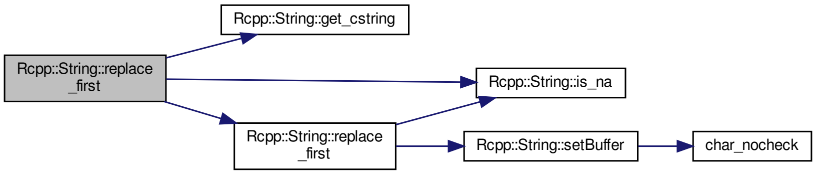
6.743.4.54 replace_first() [2/4]

```
String& Rcpp::String::replace_first (
             const char * s,
             const Rcpp::String & news )  [inline]
```

Definition at line 431 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:



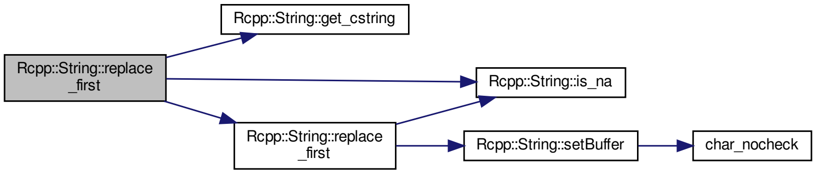
6.743.4.55 replace_first() [3/4]

```
String& Rcpp::String::replace_first (
             const Rcpp::String & s,
             const char * news )  [inline]
```

Definition at line 426 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:



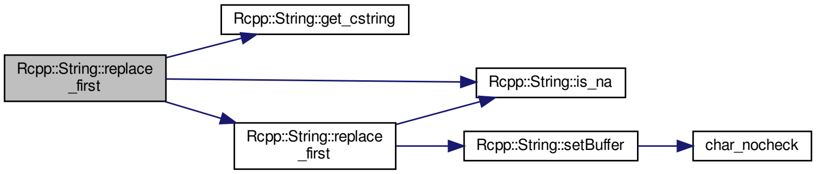
6.743.4.56 replace_first() [4/4]

```
String& Rcpp::String::replace_first (
             const Rcpp::String & s,
             const Rcpp::String & news )  [inline]
```

Definition at line 436 of file String.h.

References get_cstring(), is_na(), and replace_first().

Here is the call graph for this function:



6.743.4.57 replace_last() [1/4]

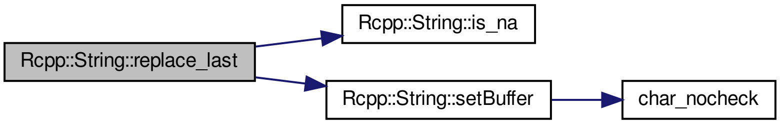
```
String& Rcpp::String::replace_last (
             const char * s,
             const char * news )  [inline]
```

Definition at line 442 of file String.h.

References buffer, is_na(), RCPP_STRING_DEBUG_2, setBuffer(), and valid.

Referenced by replace_last().

Here is the call graph for this function:



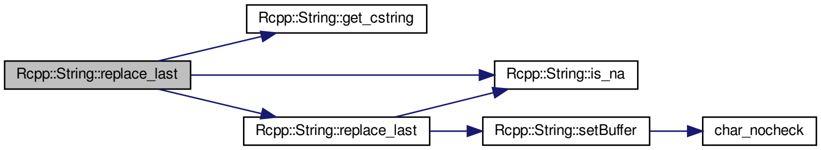
6.743.4.58 replace_last() [2/4]

```
String& Rcpp::String::replace_last (
             const char * s,
             const Rcpp::String & news )  [inline]
```

Definition at line 457 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::replace_last
Rcpp::String::is_na
Rcpp::String::replace_last
Rcpp::String::setBuffer
char_nocheck

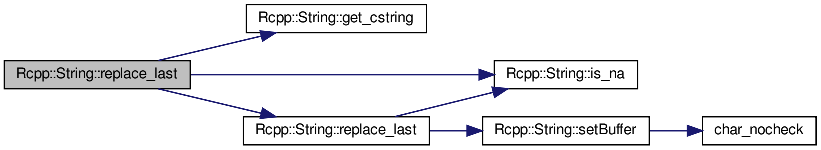
6.743.4.59 replace_last() [3/4]

```
String& Rcpp::String::replace_last (
             const Rcpp::String & s,
             const char * news )  [inline]
```

Definition at line 452 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:


Rcpp::String::get_cstring
Rcpp::String::replace_last
Rcpp::String::is_na
Rcpp::String::replace_last
Rcpp::String::setBuffer
char_nocheck

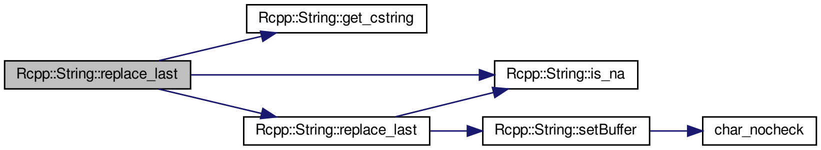
6.743.4.60 replace_last() [4/4]

```
String& Rcpp::String::replace_last (
            const Rcpp::String & s,
            const Rcpp::String & news )  [inline]
```

Definition at line 462 of file String.h.

References get_cstring(), is_na(), and replace_last().

Here is the call graph for this function:


Rcpp::String::replace_last
Rcpp::String::get_cstring
Rcpp::String::is_na
Rcpp::String::replace_last
Rcpp::String::setBuffer
char_nocheck

6.743.4.61 set_encoding()

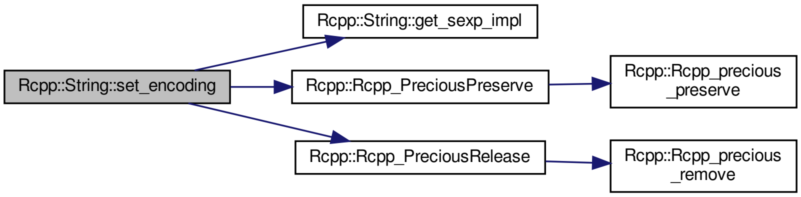
```
void Rcpp::String::set_encoding (
            cetype_t encoding )  [inline]
```

Definition at line 584 of file String.h.

References data, enc, get_sexp_impl(), Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Referenced by String().

Here is the call graph for this function:


Rcpp::String::set_encoding
Rcpp::String::get_sexp_impl
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious
_preserve
Rcpp::Rcpp_precious
_remove

6.743.4.62 set_na()

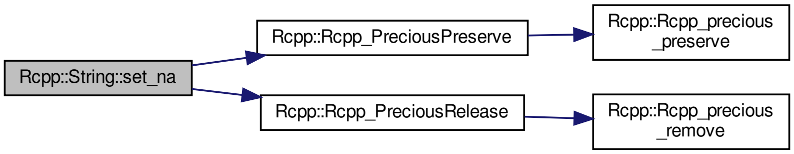
```
void Rcpp::String::set_na ( )  [inline]
```

Definition at line 535 of file String.h.

References buffer_ready, data, Rcpp::Rcpp_PreciousPreserve(), Rcpp::Rcpp_PreciousRelease(), token, and valid.

Referenced by push_back(), and push_front().

Here is the call graph for this function:


Rcpp::String::set_na
Rcpp::Rcpp_PreciousPreserve
Rcpp::Rcpp_PreciousRelease
Rcpp::Rcpp_precious_preserve
Rcpp::Rcpp_precious_remove

6.743.4.63 setBuffer()

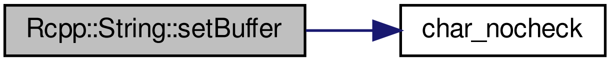
```
void Rcpp::String::setBuffer ( )  [inline], [private]
```

Definition at line 658 of file String.h.

References buffer, buffer_ready, char_nocheck(), and data.

Referenced by operator+=(), push_back(), push_front(), replace_all(), replace_first(), and replace_last().

Here is the call graph for this function:



6.743.4.64 setData()

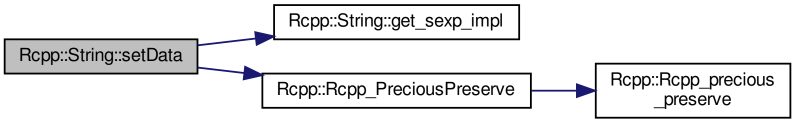
```
void Rcpp::String::setData ( )  [inline], [private]
```

Definition at line 664 of file String.h.

References data, get_sexp_impl(), Rcpp::Rcpp_PreciousPreserve(), RCPP_STRING_DEBUG, token, and valid.

Referenced by append_wide_string(), and get_sexp().

Here is the call graph for this function:



6.743.5 Member Data Documentation

6.743.5.1 buffer

```
std::string Rcpp::String::buffer  [private]
```

a buffer used to do string operations withough going back to the SEXP

Definition at line 646 of file String.h.

Referenced by append(), get_cstring(), get_sexp_impl(), operator+=(), operator=(), push_back(), push_front(), replace_all(), replace_first(), replace_last(), and setBuffer().

6.743.5.2 buffer_ready

```
bool Rcpp::String::buffer_ready  [private]
```

is the buffer initialized

Definition at line 652 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_cstring(), operator+=(), operator=(), set_na(), setBuffer(), and String().

6.743.5.3 data

```
SEXP Rcpp::String::data  [private]
```

the CHARSXP this String encapsulates

Definition at line 642 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_cstring(), get_sexp(), is_na(), operator+=(), operator=(), set_encoding(), set_na(), setBuffer(), setData(), String(), and ∼String().

6.743.5.4 enc

```
cetype_t Rcpp::String::enc  [private]
```

the encoding of encapsulated CHARSXP

Definition at line 655 of file String.h.

Referenced by get_encoding(), get_sexp_impl(), operator=(), set_encoding(), and String().

6.743.5.5 token

```
SEXP Rcpp::String::token  [private]
```

Definition at line 643 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), operator+=(), operator=(), set_encoding(), set_na(), set←Data(), String(), and ∼String().

6.743.5.6 valid

```
bool Rcpp::String::valid  [private]
```

is data in sync with buffer

Definition at line 649 of file String.h.

Referenced by append_wide_string(), assign_wide_string(), get_sexp(), operator+=(), operator=(), push_back(), push←_front(), replace_all(), replace_first(), replace_last(), set_encoding(), set_na(), setData(), and String().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/String.h

6.744 Rcpp::internal::string_element_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

Public Types

- typedef SEXP target

Static Public Member Functions

- template<typename T >
  static SEXP get (const T &input)
- static SEXP get (const std::string &input)
- static SEXP get (const Rcpp::String &input)
- static SEXP get (const char &input)
- static SEXP get (SEXP x)

6.744.1 Detailed Description

template<int RTYPE>
class Rcpp::internal::string_element_converter< RTYPE >

Definition at line 43 of file converter.h.

6.744.2 Member Typedef Documentation

6.744.2.1 target

```
template<int RTYPE>
typedef SEXP Rcpp::internal::string_element_converter< RTYPE >::target
```

Definition at line 45 of file converter.h.

6.744.3 Member Function Documentation

6.744.3.1 get() [1/5]

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
             const char & input )  [inline], [static]
```

Definition at line 61 of file converter.h.

References RCPP_DEBUG.

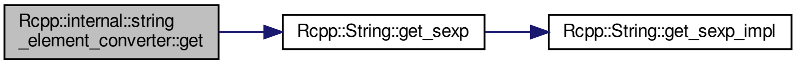
6.744.3.2 get() [2/5]

```
template<int RTYPE>
SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
             const Rcpp::String & input )  [static]
```

Definition at line 688 of file String.h.

References Rcpp::String::get_sexp(), and RCPP_DEBUG.

Here is the call graph for this function:



6.744.3.3 get() [3/5]

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
             const std::string & input )  [inline], [static]
```

Definition at line 54 of file converter.h.

References RCPP_DEBUG.

6.744.3.4 get() [4/5]

```
template<int RTYPE>
template<typename T >
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
             const T & input )  [inline], [static]
```

Definition at line 48 of file converter.h.

References DEMANGLE, and RCPP_DEBUG_1.

6.744.3.5 get() [5/5]

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
             SEXP x )  [inline], [static]
```

Definition at line 67 of file converter.h.

References RCPP_DEBUG.

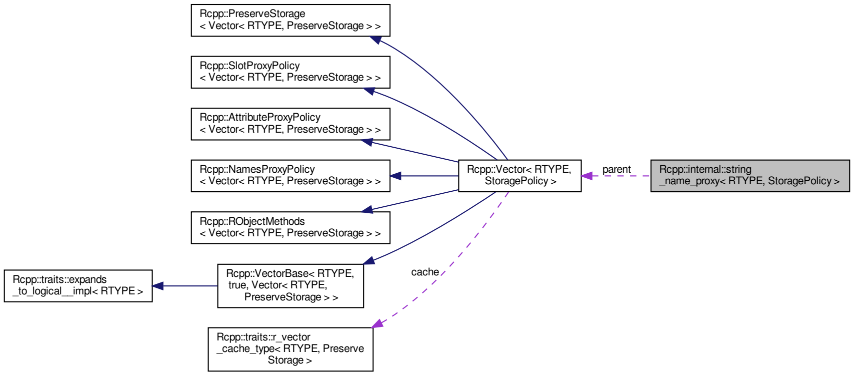
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/converter.h
- inst/include/Rcpp/String.h

6.745 Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
Rcpp::traits::r_vector
_cache_type< RTYPE, Preserve
Storage >
Rcpp::Vector< RTYPE,
StoragePolicy >
parent
cache
Rcpp::internal::string
_name_proxy< RTYPE, StoragePolicy >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef const char ∗ iterator
- typedef const char & reference

Public Member Functions

- string_name_proxy (VECTOR &v, const std::string &name_)
- string_name_proxy (const string_name_proxy &other)
- ∼string_name_proxy ()
- string_name_proxy & operator= (const std::string &rhs)
- string_name_proxy & operator= (const string_name_proxy &other)
- operator char ∗ () const
- operator SEXP () const
- iterator begin ()
- iterator end ()
- reference operator[ ] (R_xlen_t i)
- R_xlen_t size ()

Private Member Functions

- void set (const std::string &rhs)
- char ∗ get () const

Private Attributes

- VECTOR & parent
- std::string name

6.745.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >

Definition at line 84 of file proxy.h.

6.745.2 Member Typedef Documentation

6.745.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char* Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 87 of file proxy.h.

6.745.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 88 of file proxy.h.

6.745.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::string_name_proxy< RTYPE, Storage↩
Policy >::VECTOR
```

Definition at line 86 of file proxy.h.

6.745.3 Constructor & Destructor Documentation

6.745.3.1 string_name_proxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::string_name_proxy (
             VECTOR & v,
             const std::string & name_ )  [inline]
```

Definition at line 90 of file proxy.h.

6.745.3.2 string_name_proxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::string_name_proxy (
            const string_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 92 of file proxy.h.

6.745.3.3 ∼string_name_proxy()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::∼string_name_proxy ( ) [inline]
```

Definition at line 94 of file proxy.h.

6.745.4 Member Function Documentation

6.745.4.1 begin()

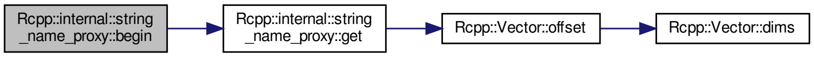
```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::begin ( ) [inline]
```

Definition at line 113 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get().

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



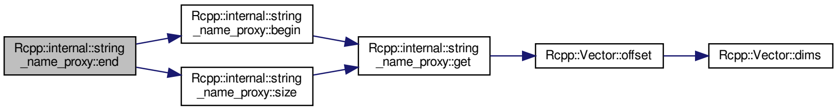
6.745.4.2 end()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::end ( ) [inline]
```

Definition at line 114 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::internal::string_name_proxy::end
Rcpp::internal::string_name_proxy::begin
Rcpp::internal::string_name_proxy::size
Rcpp::internal::string_name_proxy::get
Rcpp::Vector::offset
Rcpp::Vector::dims

6.745.4.3 get()

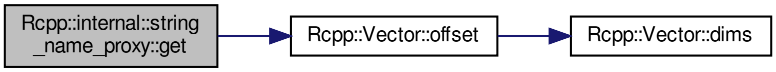
```
template<int RTYPE, template< class > class StoragePolicy>
char* Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get ( ) const [inline], [private]
```

Definition at line 130 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator char ∗(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator[ ](), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



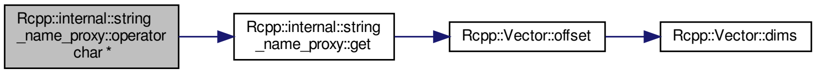
6.745.4.4 operator char ∗()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator char ∗ ( ) const  [inline]
```

Definition at line 105 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



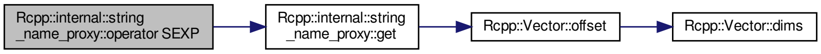
6.745.4.5 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

Definition at line 109 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



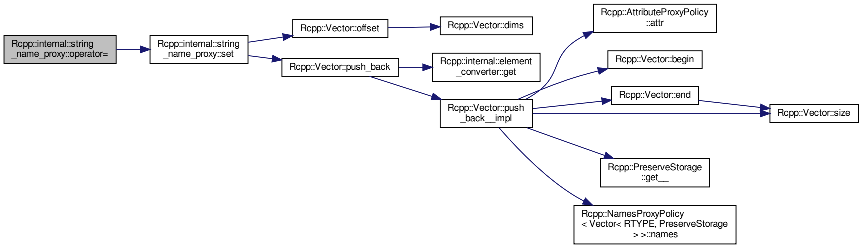
6.745.4.6 operator=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator= (
            const std::string & rhs )  [inline]
```

Definition at line 96 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string_name_proxy::operator=
Rcpp::internal::string_name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::dims
Rcpp::Vector::push_back
Rcpp::internal::element_converter::get
Rcpp::Vector::push__back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

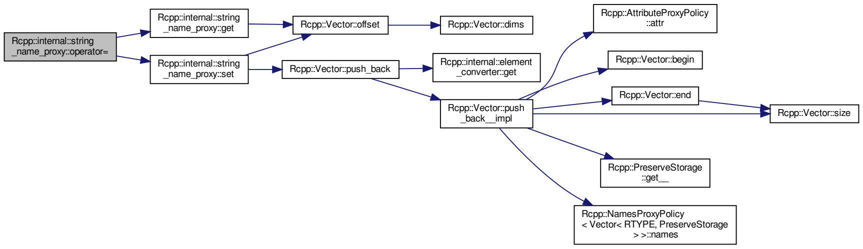
6.745.4.7 operator=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator= (
            const string_name_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 100 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string_name_proxy::operator=
Rcpp::internal::string_name_proxy::get
Rcpp::internal::string_name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::dims
Rcpp::Vector::push_back
Rcpp::internal::element_converter::get
Rcpp::Vector::push__back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

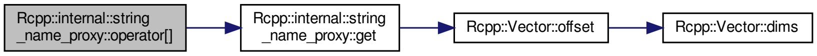
6.745.4.8 operator[]()

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator[] (
            R_xlen_t i ) [inline]
```

Definition at line 115 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:


Rcpp::internal::string _name_proxy::operator[]
Rcpp::internal::string _name_proxy::get
Rcpp::Vector::offset
Rcpp::Vector::dims

6.745.4.9 set()

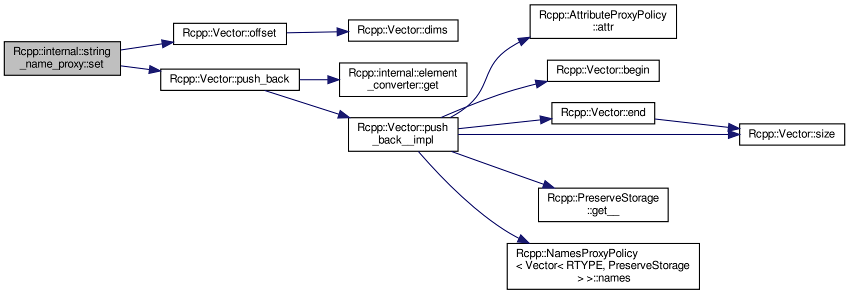
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set (
            const std::string & rhs ) [inline], [private]
```

Definition at line 121 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::parent, and Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:


Rcpp::internal::string _name_proxy::set
Rcpp::Vector::offset
Rcpp::Vector::push_back
Rcpp::Vector::dims
Rcpp::internal::element _converter::get
Rcpp::Vector::push _back__impl
Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.745.4.10 size()

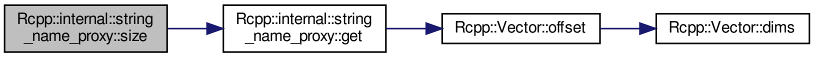
```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::size ( )  [inline]
```

Definition at line 116 of file proxy.h.

References Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get().

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



6.745.5 Member Data Documentation

6.745.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::name  [private]
```

Definition at line 120 of file proxy.h.

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set().

6.745.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::parent  [private]
```

Definition at line 119 of file proxy.h.

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set().

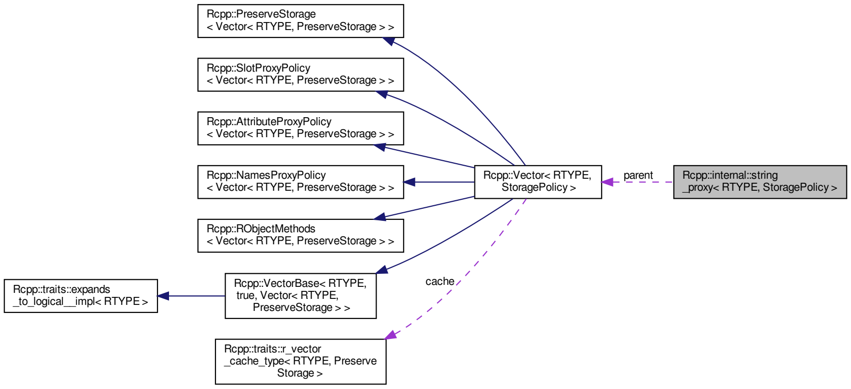
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/proxy.h

6.746 Rcpp::internal::string_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <string_proxy.h>
```

Collaboration diagram for Rcpp::internal::string_proxy< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::traits::r_vector _cache_type< RTYPE, Preserve Storage >
Rcpp::Vector< RTYPE, StoragePolicy >
cache
parent
Rcpp::internal::string _proxy< RTYPE, StoragePolicy >

Public Types

- typedef ::Rcpp::Vector< RTYPE, StoragePolicy > VECTOR
- typedef const char ∗ iterator
- typedef const char & reference

Public Member Functions

- string_proxy ()
- string_proxy (VECTOR &v, R_xlen_t index_)
- string_proxy (const string_proxy &other)
- string_proxy & operator= (const string_proxy< RTYPE, StoragePolicy > &other)
- template<template< class > class StoragePolicy2>
  string_proxy & operator= (const string_proxy< RTYPE, StoragePolicy2 > &other)
- template<template< class > class StoragePolicy2>
  string_proxy & operator= (const const_string_proxy< RTYPE, StoragePolicy2 > &other)
- string_proxy & operator= (const String &s)
- template<typename T >
  string_proxy & operator= (const std::basic_string< T > &rhs)
- string_proxy & operator= (const char ∗rhs)
- string_proxy & operator= (const wchar_t ∗rhs)
- string_proxy & operator= (SEXP rhs)
- void import (const string_proxy &other)
- template<typename T >
  string_proxy & operator+= (const T &rhs)

- operator SEXP () const
- operator char ∗ () const
- void swap (string_proxy &other)
- void move (R_xlen_t n)
- SEXP get () const
- template<typename T >
  void set (const T &x)
- void set (SEXP x)
- iterator begin () const
- iterator end () const
- R_xlen_t size () const
- bool empty () const
- reference operator[ ] (R_xlen_t n)
- template<typename UnaryOperator >
  void transform (UnaryOperator op)
- template<typename OutputIterator , typename UnaryOperator >
  void apply (OutputIterator target, UnaryOperator op)
- template<typename UnaryOperator >
  void apply (UnaryOperator op)
- bool operator== (const char ∗other) const
- bool operator!= (const char ∗other) const
- template<template< class > class SP>
  bool operator== (const string_proxy< STRSXP, SP > &other) const
- template<template< class > class SP>
  bool operator!= (const string_proxy< STRSXP, SP > &other) const
- bool operator== (SEXP other) const
- bool operator!= (SEXP other) const
- template<typename T >
  string_proxy< RTYPE, StoragePolicy > & operator+= (const T &rhs)
- template<template< class > class StoragePolicy2>
  string_proxy< RTYPE, StoragePolicy1 > & operator= (const const_string_proxy< RTYPE, StoragePolicy2 > &other)

Public Attributes

- VECTOR ∗ parent
- R_xlen_t index

Static Private Attributes

- static std::string buffer

Friends

- template<int RT>
  std::ostream & operator<< (std::ostream &os, const string_proxy< RT > &proxy)
- template<int RT>
  std::string operator+ (const std::string &x, const string_proxy< RT > &proxy)

6.746.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::string_proxy< RTYPE, StoragePolicy >

Definition at line 29 of file string_proxy.h.

6.746.2 Member Typedef Documentation

6.746.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char* Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 33 of file string_proxy.h.

6.746.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 34 of file string_proxy.h.

6.746.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::string_proxy< RTYPE, StoragePolicy
>::VECTOR
```

Definition at line 32 of file string_proxy.h.

6.746.3 Constructor & Destructor Documentation

6.746.3.1 string_proxy() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy ( )  [inline]
```

Definition at line 36 of file string_proxy.h.

6.746.3.2 string_proxy() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy (
             VECTOR & v,
             R_xlen_t index_ )  [inline]
```

Creates a proxy

Parameters

v	reference to the associated character vector
index	index

Definition at line 44 of file string_proxy.h.

6.746.3.3 string_proxy() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy (
             const string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 46 of file string_proxy.h.

6.746.4 Member Function Documentation

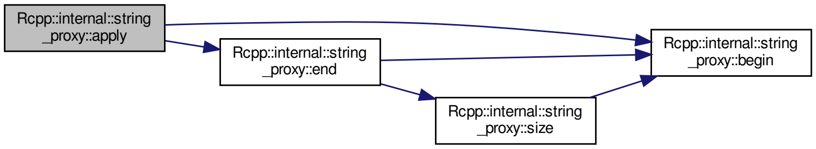
6.746.4.1 apply() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename OutputIterator , typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply (
             OutputIterator target,
             UnaryOperator op )  [inline]
```

Definition at line 181 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



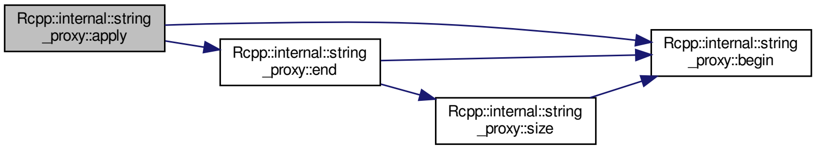
6.746.4.2 apply() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply (
             UnaryOperator op )  [inline]
```

Definition at line 186 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



6.746.4.3 begin()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin ( ) const  [inline]
```

Definition at line 167 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::empty(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!=(), Rcpp::internal::operator<(), Rcpp::internal::operator<=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator==(), Rcpp::internal::operator>(), Rcpp::internal::operator>=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator[ ](), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::size(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::transform().

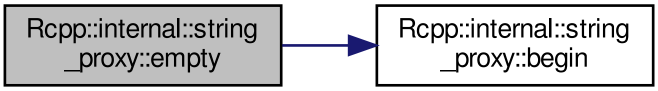
6.746.4.4 empty()

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::empty ( ) const  [inline]
```

Definition at line 170 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.746.4.5 end()

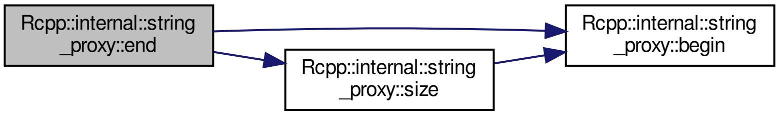
```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end ( ) const  [inline]
```

Definition at line 168 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply().

Here is the call graph for this function:



6.746.4.6 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get ( ) const  [inline]
```

Definition at line 156 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator char ∗(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::String::operator!=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!=(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::String::operator=(), Rcpp::String::operator==(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator==().

6.746.4.7 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::import (
            const string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 102 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent.

6.746.4.8 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::move (
             R_xlen_t n )  [inline]
```

Definition at line 154 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index.

6.746.4.9 operator char *()

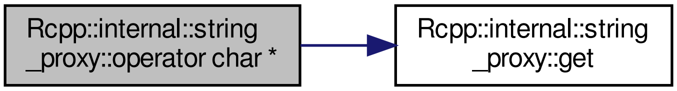
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator char * ( ) const  [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to and convert it to a C string

Definition at line 127 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



6.746.4.10 operator SEXP()

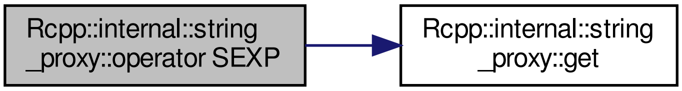
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const  [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to.

Definition at line 118 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



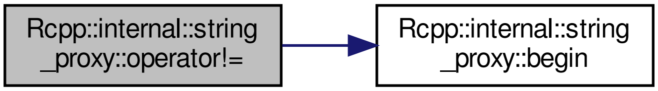
6.746.4.11 operator"!=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
            const char * other ) const  [inline]
```

Definition at line 193 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



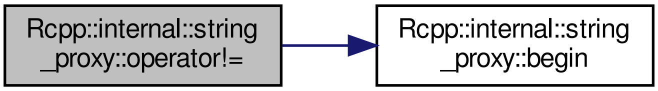
6.746.4.12 operator"!=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class SP>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
            const string_proxy< STRSXP, SP > & other ) const  [inline]
```

Definition at line 203 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



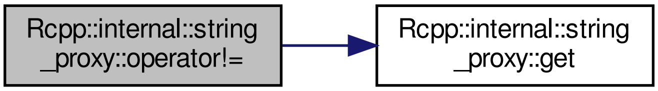
6.746.4.13 operator"!=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
             SEXP other ) const  [inline]
```

Definition at line 211 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



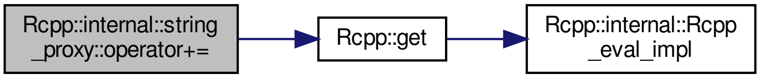
6.746.4.14 operator+=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy<RTYPE, StoragePolicy>& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator+=
(
             const T & rhs )
```

Definition at line 700 of file String.h.

References Rcpp::get().

Here is the call graph for this function:



6.746.4.15 operator+=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator+= (
             const T & rhs )
```

lhs use. Adds the content of the rhs proxy to the element this proxy refers to.

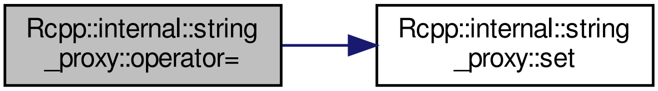
6.746.4.16 operator=() [1/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const char * rhs )  [inline]
```

Definition at line 85 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



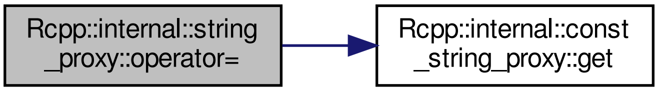
6.746.4.17 operator=() [2/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy<RTYPE, StoragePolicy1>& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >↩
::operator= (
             const const_string_proxy< RTYPE, StoragePolicy2 > & other )
```

Definition at line 174 of file const_string_proxy.h.

References Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



6.746.4.18 operator=() [3/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const const_string_proxy< RTYPE, StoragePolicy2 > & other )
```

6.746.4.19 operator=() [4/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const std::basic_string< T > & rhs )  [inline]
```

lhs use. Assigns the value of the referred element of the character vector

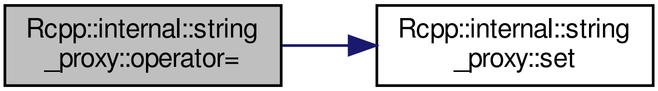
Parameters

rhs	new content for the element referred by this proxy

Definition at line 80 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



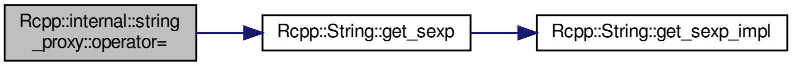
6.746.4.20 operator=() [5/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy< RTYPE, StoragePolicy > & Rcpp::internal::string_proxy< RTYPE, StoragePolicy >↵
::operator= (
             const String & s )
```

Definition at line 682 of file String.h.

References Rcpp::String::get_sexp().

Here is the call graph for this function:



6.746.4.21 operator=() [6/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

lhs use. Assign the value of the referred element to the current content of the element referred by the rhs proxy

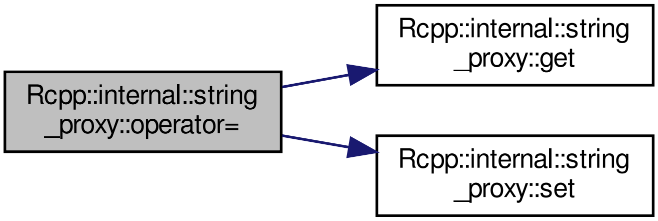
Parameters

rhs	another proxy, possibly from another vector

Definition at line 57 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string_proxy::operator=
Rcpp::internal::string_proxy::get
Rcpp::internal::string_proxy::set

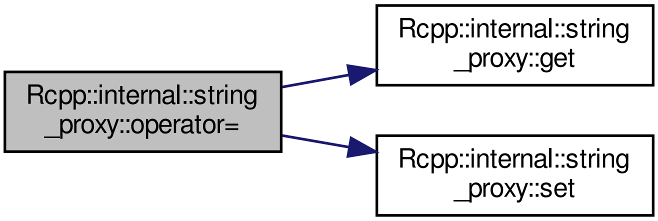
6.746.4.22 operator=() [7/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const string_proxy< RTYPE, StoragePolicy2 > & other )  [inline]
```

Definition at line 63 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string_proxy::operator=
Rcpp::internal::string_proxy::get
Rcpp::internal::string_proxy::set

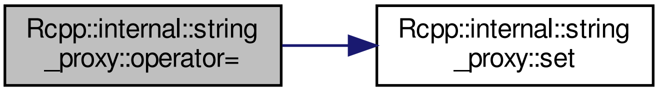
6.746.4.23 operator=() [8/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
             const wchar_t * rhs ) [inline]
```

Definition at line 90 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string_proxy::operator=
Rcpp::internal::string_proxy::set

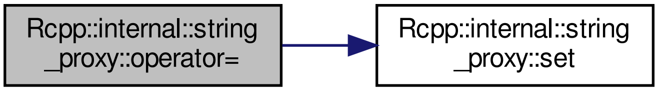
6.746.4.24 operator=() [9/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
            SEXP rhs )  [inline]
```

Definition at line 96 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



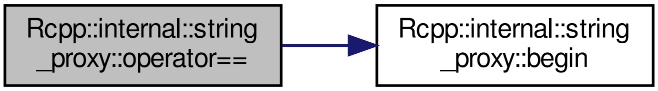
6.746.4.25 operator==() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator== (
            const char * other ) const  [inline]
```

Definition at line 190 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



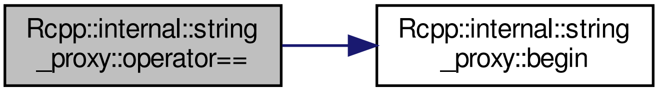
6.746.4.26 operator==() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class SP>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator== (
            const string_proxy< STRSXP, SP > & other ) const  [inline]
```

Definition at line 198 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



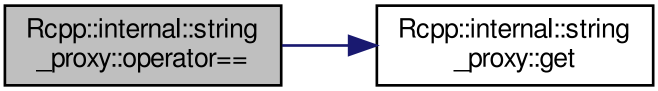
6.746.4.27 operator==() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator== (
            SEXP other ) const  [inline]
```

Definition at line 207 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



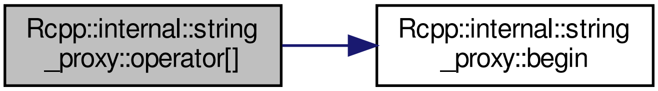
6.746.4.28 operator[]()

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator[] (
            R_xlen_t n ) [inline]
```

Definition at line 171 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



6.746.4.29 set() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set (
            const T & x ) [inline]
```

Definition at line 160 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator=(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::transform().

6.746.4.30 set() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set (
            SEXP x ) [inline]
```

Definition at line 163 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent.

6.746.4.31 size()

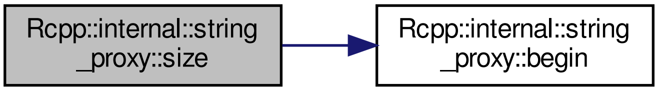
```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::size ( ) const  [inline]
```

Definition at line 169 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin().

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



6.746.4.32 swap()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::swap (
             string_proxy< RTYPE, StoragePolicy > & other )  [inline]
```

Definition at line 146 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent.

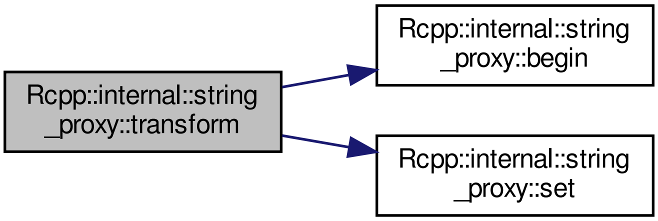
6.746.4.33 transform()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::transform (
             UnaryOperator op )  [inline]
```

Definition at line 174 of file string_proxy.h.

References Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::buffer, and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:


Rcpp::internal::string _proxy::transform
Rcpp::internal::string _proxy::begin
Rcpp::internal::string _proxy::set

6.746.5 Friends And Related Function Documentation

6.746.5.1 operator+

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT>
std::string operator+ (
             const std::string & x,
             const string_proxy< RT > & proxy )  [friend]
```

6.746.5.2 operator<<

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT>
std::ostream& operator<< (
             std::ostream & os,
             const string_proxy< RT > & proxy )  [friend]
```

Prints the element this proxy refers to to an output stream

6.746.6 Member Data Documentation

6.746.6.1 buffer

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::buffer  [static], [private]
```

Definition at line 216 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::transform().

6.746.6.2 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 153 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::import(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::move(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal↩ ::string_proxy< RTYPE, StoragePolicy >::swap().

6.746.6.3 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR* Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 152 of file string_proxy.h.

Referenced by Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::import(), Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::swap().

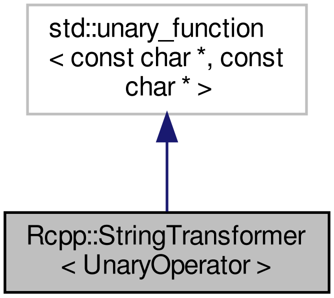
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_proxy.h
- inst/include/Rcpp/vector/string_proxy.h
- inst/include/Rcpp/String.h
- inst/include/Rcpp/vector/const_string_proxy.h

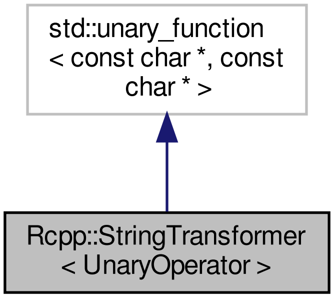
6.747 Rcpp::StringTransformer< UnaryOperator > Class Template Reference

```
#include <StringTransformer.h>
```

Inheritance diagram for Rcpp::StringTransformer< UnaryOperator >:



Collaboration diagram for Rcpp::StringTransformer< UnaryOperator >:



Public Member Functions

- StringTransformer (const UnaryOperator &op_)
- ∼StringTransformer ()
- const char ∗ operator() (const char ∗input)

Private Attributes

- const UnaryOperator & op
- std::string buffer

6.747.1 Detailed Description

template<typename UnaryOperator>
class Rcpp::StringTransformer< UnaryOperator >

Definition at line 30 of file StringTransformer.h.

6.747.2 Constructor & Destructor Documentation

6.747.2.1 StringTransformer()

```
template<typename UnaryOperator >
Rcpp::StringTransformer< UnaryOperator >::StringTransformer (
            const UnaryOperator & op_ )  [inline]
```

Definition at line 35 of file StringTransformer.h.

6.747.2.2 ∼StringTransformer()

```
template<typename UnaryOperator >
Rcpp::StringTransformer< UnaryOperator >::∼StringTransformer ( )  [inline]
```

Definition at line 36 of file StringTransformer.h.

6.747.3 Member Function Documentation

6.747.3.1 operator()()

```
template<typename UnaryOperator >
const char* Rcpp::StringTransformer< UnaryOperator >::operator() (
            const char * input )  [inline]
```

Definition at line 38 of file StringTransformer.h.

References Rcpp::StringTransformer< UnaryOperator >::buffer, and Rcpp::StringTransformer< UnaryOperator >::op.

6.747.4 Member Data Documentation

6.747.4.1 buffer

```
template<typename UnaryOperator >
std::string Rcpp::StringTransformer< UnaryOperator >::buffer  [private]
```

Definition at line 46 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

6.747.4.2 op

```
template<typename UnaryOperator >
const UnaryOperator& Rcpp::StringTransformer< UnaryOperator >::op  [private]
```

Definition at line 45 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

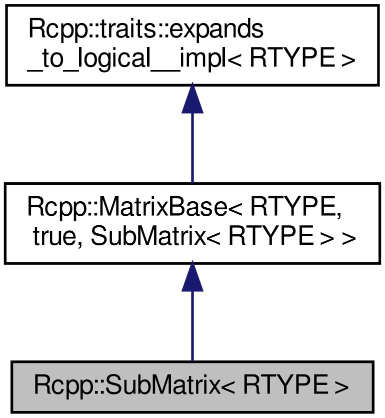
The documentation for this class was generated from the following file:

- inst/include/Rcpp/StringTransformer.h

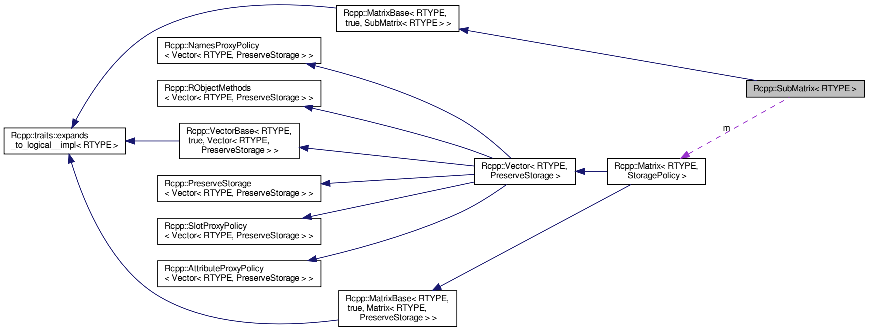
6.748 Rcpp::SubMatrix< RTYPE > Class Template Reference

```
#include <SubMatrix.h>
```

Inheritance diagram for Rcpp::SubMatrix< RTYPE >:



Collaboration diagram for Rcpp::SubMatrix< RTYPE >:


Rcpp::MatrixBase< RTYPE, true, SubMatrix< RTYPE > >
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods < Vector< RTYPE, PreserveStorage > >
Rcpp::SubMatrix< RTYPE >
m
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::Vector< RTYPE, PreserveStorage >
Rcpp::Matrix< RTYPE, StoragePolicy >
Rcpp::PreserveStorage < Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy < Vector< RTYPE, PreserveStorage > >
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >

Public Types

- typedef Matrix< RTYPE > MATRIX
- typedef Vector< RTYPE >::iterator vec_iterator
- typedef MATRIX::Proxy Proxy

Public Member Functions

- SubMatrix (MATRIX &m_, const Range &row_range_, const Range &col_range_)
- R_xlen_t size () const
- int ncol () const
- int nrow () const
- Proxy operator() (int i, int j) const
- vec_iterator column_iterator (int j) const

Private Attributes

- MATRIX & m
- vec_iterator iter
- int m_nr
- int nc
- int nr

Additional Inherited Members

6.748.1 Detailed Description

template<int RTYPE>
class Rcpp::SubMatrix< RTYPE >

Definition at line 28 of file SubMatrix.h.

6.748.2 Member Typedef Documentation

6.748.2.1 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::SubMatrix< RTYPE >::MATRIX
```

Definition at line 30 of file SubMatrix.h.

6.748.2.2 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::SubMatrix< RTYPE >::Proxy
```

Definition at line 32 of file SubMatrix.h.

6.748.2.3 vec_iterator

```
template<int RTYPE>
typedef Vector<RTYPE>::iterator Rcpp::SubMatrix< RTYPE >::vec_iterator
```

Definition at line 31 of file SubMatrix.h.

6.748.3 Constructor & Destructor Documentation

6.748.3.1 SubMatrix()

```
template<int RTYPE>
Rcpp::SubMatrix< RTYPE >::SubMatrix (
            MATRIX & m_,
            const Range & row_range_,
            const Range & col_range_ )  [inline]
```

Definition at line 34 of file SubMatrix.h.

6.748.4 Member Function Documentation

6.748.4.1 column_iterator()

```
template<int RTYPE>
vec_iterator Rcpp::SubMatrix< RTYPE >::column_iterator (
             int j ) const  [inline]
```

Definition at line 50 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::iter, and Rcpp::SubMatrix< RTYPE >::m_nr.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Matrix< RTYPE, StoragePolicy >↩ ::operator=().

6.748.4.2 ncol()

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::ncol ( ) const  [inline]
```

Definition at line 43 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::nc.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::SubMatrix< RTYPE >::size().

6.748.4.3 nrow()

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nrow ( ) const  [inline]
```

Definition at line 44 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::nr.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::SubMatrix< RTYPE >::size().

6.748.4.4 operator()()

```
template<int RTYPE>
Proxy Rcpp::SubMatrix< RTYPE >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 46 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::iter, and Rcpp::SubMatrix< RTYPE >::m_nr.

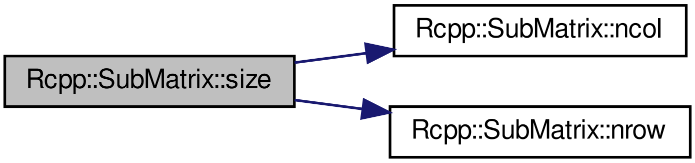
6.748.4.5 size()

```
template<int RTYPE>
R_xlen_t Rcpp::SubMatrix< RTYPE >::size ( ) const  [inline]
```

Definition at line 42 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::ncol(), and Rcpp::SubMatrix< RTYPE >::nrow().

Here is the call graph for this function:



6.748.5 Member Data Documentation

6.748.5.1 iter

```
template<int RTYPE>
vec_iterator Rcpp::SubMatrix< RTYPE >::iter  [private]
```

Definition at line 54 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::column_iterator(), and Rcpp::SubMatrix< RTYPE >::operator()().

6.748.5.2 m

```
template<int RTYPE>
MATRIX& Rcpp::SubMatrix< RTYPE >::m  [private]
```

Definition at line 53 of file SubMatrix.h.

6.748.5.3 m_nr

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::m_nr  [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::column_iterator(), and Rcpp::SubMatrix< RTYPE >::operator()().

6.748.5.4 nc

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nc  [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::ncol().

6.748.5.5 nr

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nr  [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::nrow().

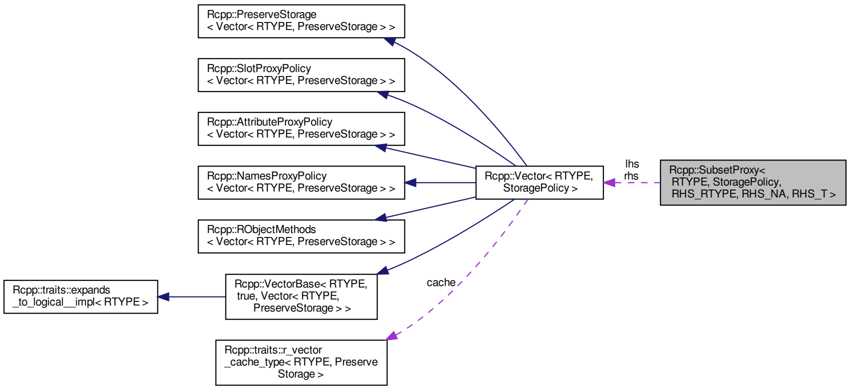
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/SubMatrix.h

6.749 Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > Class Template Reference

```
#include <Subsetter.h>
```

Collaboration diagram for Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE,
true, Vector< RTYPE,
PreserveStorage > >
Rcpp::traits::r_vector
_cache_type< RTYPE, Preserve
Storage >
Rcpp::Vector< RTYPE,
StoragePolicy >
cache
lhs
rhs
Rcpp::SubsetProxy<
RTYPE, StoragePolicy,
RHS_RTYPE, RHS_NA, RHS_T >

Public Member Functions

- SubsetProxy (LHS_t &lhs_, const RHS_t &rhs_)
- SubsetProxy (const SubsetProxy &other)
- template<int OtherRTYPE, template< class > class OtherStoragePolicy>
  SubsetProxy & operator= (const Vector< OtherRTYPE, OtherStoragePolicy > &other)
- SubsetProxy & operator= (double other)
- SubsetProxy & operator= (int other)
- SubsetProxy & operator= (const char ∗other)
- SubsetProxy & operator= (bool other)
- template<int RTYPE_OTHER, template< class > class StoragePolicyOther, int RHS_RTYPE_OTHER, bool RHS_NA_OTHER, typename RHS_T_OTHER >
  SubsetProxy & operator= (const SubsetProxy< RTYPE_OTHER, StoragePolicyOther, RHS_RTYPE_OTHER, RHS_NA_OTHER, RHS_T_OTHER > &other)
- SubsetProxy & operator= (const SubsetProxy &other)
- operator Vector< RTYPE, StoragePolicy > () const
- operator SEXP () const

Private Types

- typedef Vector< RTYPE, StoragePolicy > LHS_t
- typedef Vector< RHS_RTYPE, StoragePolicy > RHS_t

Private Member Functions

- template<typename IDX >
  void check_indices (IDX ∗x, R_xlen_t n, R_xlen_t size)
- void get_indices (traits::identity< traits::int2type< INTSXP > > t)
- void get_indices (traits::identity< traits::int2type< REALSXP > > t)
- void get_indices (traits::identity< traits::int2type< STRSXP > > t)
- void get_indices (traits::identity< traits::int2type< LGLSXP > > t)
- Vector< RTYPE, StoragePolicy > get_vec () const

Private Attributes

- LHS_t & lhs
- const RHS_t & rhs
- R_xlen_t lhs_n
- R_xlen_t rhs_n
- std::vector< R_xlen_t > indices
- R_xlen_t indices_n

6.749.1 Detailed Description

**template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T>
class Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >**

Definition at line 33 of file Subsetter.h.

6.749.2 Member Typedef Documentation

6.749.2.1 LHS_t

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
typedef Vector<RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA,
RHS_T >::LHS_t [private]
```

Definition at line 35 of file Subsetter.h.

6.749.2.2 RHS_t

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
typedef Vector<RHS_RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE,
RHS_NA, RHS_T >::RHS_t  [private]
```

Definition at line 36 of file Subsetter.h.

6.749.3 Constructor & Destructor Documentation

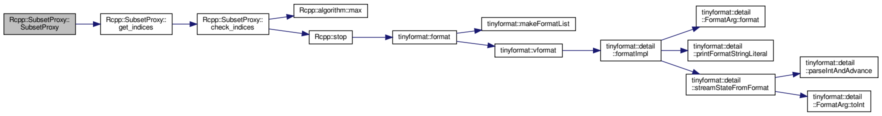
6.749.3.1 SubsetProxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::SubsetProxy (
            LHS_t & lhs_,
            const RHS_t & rhs_ )  [inline]
```

Definition at line 40 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

Here is the call graph for this function:



6.749.3.2 SubsetProxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::SubsetProxy (
            const SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > & other )  [inline]
```

Definition at line 45 of file Subsetter.h.

6.749.4 Member Function Documentation

6.749.4.1 check_indices()

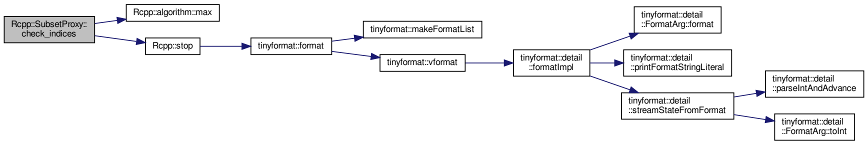
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
template<typename IDX >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices (
            IDX * x,
            R_xlen_t n,
            R_xlen_t size )  [inline], [private]
```

Definition at line 138 of file Subsetter.h.

References Rcpp::algorithm::max(), and Rcpp::stop().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

Here is the call graph for this function:



6.749.4.2 get_indices() [1/4]

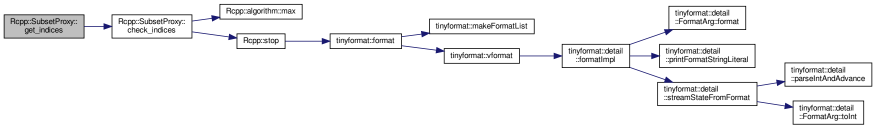
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
            traits::identity< traits::int2type< INTSXP > > t )  [inline], [private]
```

Definition at line 153 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:


Rcpp::SubsetProxy::get_indices
Rcpp::SubsetProxy::check_indices
Rcpp::algorithm::max
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

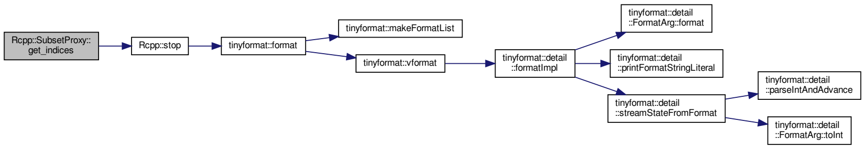
6.749.4.3 get_indices() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
            traits::identity< traits::int2type< LGLSXP > > t )  [inline], [private]
```

Definition at line 191 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n, and Rcpp::stop().

Here is the call graph for this function:


Rcpp::SubsetProxy::get_indices
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

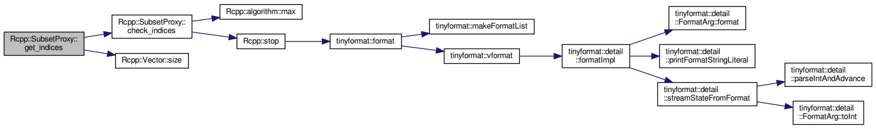
6.749.4.4 get_indices() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
            traits::identity< traits::int2type< REALSXP > > t )  [inline], [private]
```

Definition at line 163 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::SubsetProxy::get_indices
Rcpp::SubsetProxy::check_indices
Rcpp::Vector::size
Rcpp::algorithm::max
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

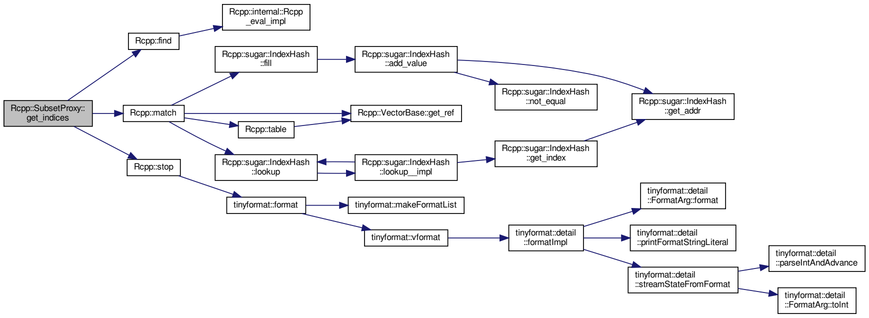
6.749.4.5 get_indices() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
            traits::identity< traits::int2type< STRSXP > > t )  [inline], [private]
```

Definition at line 176 of file Subsetter.h.

References Rcpp::find(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n, Rcpp::match(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n, and Rcpp::stop().

Here is the call graph for this function:


Rcpp::SubsetProxy::get_indices
Rcpp::find
Rcpp::internal::Rcpp_eval_impl
Rcpp::match
Rcpp::sugar::IndexHash::fill
Rcpp::sugar::IndexHash::add_value
Rcpp::sugar::IndexHash::not_equal
Rcpp::sugar::IndexHash::get_addr
Rcpp::VectorBase::get_ref
Rcpp::table
Rcpp::sugar::IndexHash::lookup
Rcpp::sugar::IndexHash::lookup__impl
Rcpp::sugar::IndexHash::get_index
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

6.749.4.6 get_vec()

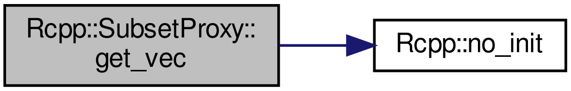
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Vector<RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_↩
T >::get_vec ( ) const  [inline], [private]
```

Definition at line 208 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::Subset↩Proxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, Storage↩Policy, 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, Storage↩Policy >().

Here is the call graph for this function:



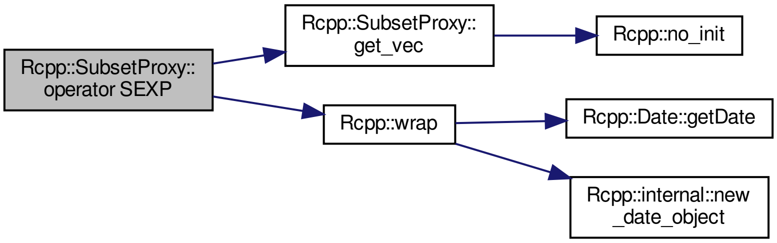
6.749.4.7 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator SEXP ( ) const
[inline]
```

Definition at line 130 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec(), and Rcpp↩::wrap().

Here is the call graph for this function:



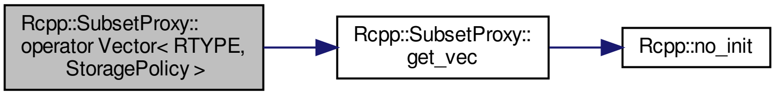
6.749.4.8 operator Vector< RTYPE, StoragePolicy >()

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator Vector< RTYPE,
StoragePolicy > ( ) const [inline]
```

Definition at line 126 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec().

Here is the call graph for this function:



6.749.4.9 operator=() [1/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             bool other )  [inline]
```

Definition at line 95 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs.

6.749.4.10 operator=() [2/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             const char * other )  [inline]
```

Definition at line 88 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs.

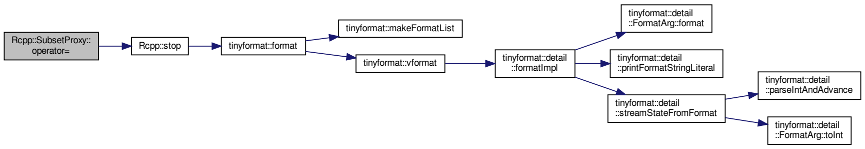
6.749.4.11 operator=() [3/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             const SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > & other )  [inline]
```

Definition at line 110 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs, and Rcpp::stop().

Here is the call graph for this function:



6.749.4.12 operator=() [4/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
template<int RTYPE_OTHER, template< class > class StoragePolicyOther, int RHS_RTYPE_OTHER, bool
RHS_NA_OTHER, typename RHS_T_OTHER >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             const SubsetProxy< RTYPE_OTHER, StoragePolicyOther, RHS_RTYPE_OTHER, RHS_NA_OTHER,
RHS_T_OTHER > & other )  [inline]
```

Definition at line 103 of file Subsetter.h.

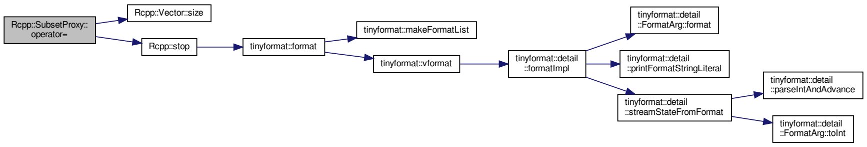
6.749.4.13 operator=() [5/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
template<int OtherRTYPE, template< class > class OtherStoragePolicy>
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             const Vector< OtherRTYPE, OtherStoragePolicy > & other )  [inline]
```

Definition at line 55 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs, Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::stop().

Here is the call graph for this function:



6.749.4.14 operator=() [6/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
             double other )  [inline]
```

Definition at line 74 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs.

6.749.4.15 operator=() [7/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
              int other )  [inline]
```

Definition at line 81 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices, Rcpp::Subset↩Proxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs.

6.749.5 Member Data Documentation

6.749.5.1 indices

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
std::vector<R_xlen_t> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >↩
::indices  [private]
```

Definition at line 230 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), Rcpp↩::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=().

6.749.5.2 indices_n

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n  [private]
```

Definition at line 233 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), Rcpp↩::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=().

6.749.5.3 lhs

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
LHS_t& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 225 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), Rcpp↩ ::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=().

6.749.5.4 lhs_n

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n  [private]
```

Definition at line 227 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

6.749.5.5 rhs

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
const RHS_t& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 226 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

6.749.5.6 rhs_n

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n  [private]
```

Definition at line 228 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/Subsetter.h

6.750 tinyformat::detail::is_convertible< T1, T2 >::succeed Struct Reference

Public Attributes

- char dummy

6.750.1 Detailed Description

template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::succeed

Definition at line 181 of file tinyformat.h.

6.750.2 Member Data Documentation

6.750.2.1 dummy

```
template<typename T1 , typename T2 >
char tinyformat::detail::is_convertible< T1, T2 >::succeed::dummy
```

Definition at line 181 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/tinyformat.h

6.751 Rcpp::sugar::sugar_const_iterator_type< T > Struct Template Reference

```
#include <iterator.h>
```

Public Types

- typedef SugarIterator< T > type

6.751.1 Detailed Description

template<typename T>
struct Rcpp::sugar::sugar_const_iterator_type< T >

Definition at line 110 of file iterator.h.

6.751.2 Member Typedef Documentation

6.751.2.1 type

```
template<typename T >
typedef SugarIterator<T> Rcpp::sugar::sugar_const_iterator_type< T >::type
```

Definition at line 111 of file iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

6.752 Rcpp::sugar::sugar_const_iterator_type< CharacterVector > Struct Reference

```
#include <iterator.h>
```

Public Types

- typedef SEXP ∗ type

6.752.1 Detailed Description

Definition at line 116 of file iterator.h.

6.752.2 Member Typedef Documentation

6.752.2.1 type

```
typedef SEXP* Rcpp::sugar::sugar_const_iterator_type< CharacterVector >::type
```

Definition at line 117 of file iterator.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

6.753 Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > > Struct Template Reference

```
#include <iterator.h>
```

Public Types

- typedef Rcpp::Vector< RTYPE >::const_iterator type

6.753.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > >

Definition at line 113 of file iterator.h.

6.753.2 Member Typedef Documentation

6.753.2.1 type

```
template<int RTYPE>
typedef Rcpp::Vector<RTYPE>::const_iterator Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector<
RTYPE > >::type
```

Definition at line 114 of file iterator.h.

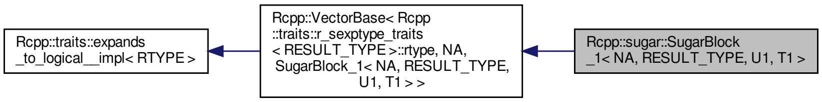
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

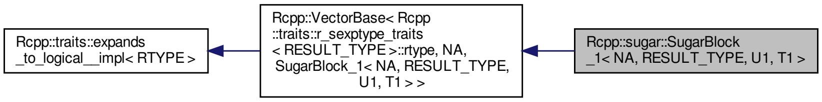
6.754 Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 > Class Template Reference

`#include <SugarBlock_1.h>`

Inheritance diagram for Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp
::traits::r_sexptype_traits
< RESULT_TYPE >::rtype, NA,
SugarBlock_1< NA, RESULT_TYPE,
U1, T1 > >
Rcpp::sugar::SugarBlock
_1< NA, RESULT_TYPE, U1, T1 >

Collaboration diagram for Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp
::traits::r_sexptype_traits
< RESULT_TYPE >::rtype, NA,
SugarBlock_1< NA, RESULT_TYPE,
U1, T1 > >
Rcpp::sugar::SugarBlock
_1< NA, RESULT_TYPE, U1, T1 >

Public Types

- typedef RESULT_TYPE(∗ FunPtr) (U1)

Public Member Functions

- SugarBlock_1 (FunPtr ptr_, const T1 &vec_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const T1 & vec

6.754.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename T1>
class Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >

Definition at line 29 of file SugarBlock_1.h.

6.754.2 Member Typedef Documentation

6.754.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::FunPtr) (U1)
```

Definition at line 31 of file SugarBlock_1.h.

6.754.3 Constructor & Destructor Documentation

6.754.3.1 SugarBlock_1()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::SugarBlock_1 (
            FunPtr ptr_,
            const T1 & vec_ )  [inline]
```

Definition at line 32 of file SugarBlock_1.h.

6.754.4 Member Function Documentation

6.754.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
RESULT_TYPE Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 34 of file SugarBlock_1.h.

References Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::ptr, and Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::vec.

6.754.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
R_xlen_t Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::size ( ) const  [inline]
```

Definition at line 37 of file SugarBlock_1.h.

References Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::vec.

6.754.5 Member Data Documentation

6.754.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
FunPtr Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::ptr  [private]
```

Definition at line 40 of file SugarBlock_1.h.

Referenced by Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::operator[ ]().

6.754.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
const T1& Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::vec  [private]
```

Definition at line 41 of file SugarBlock_1.h.

Referenced by Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::operator[ ](), and Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::size().

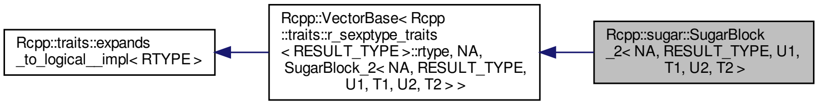
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_1.h

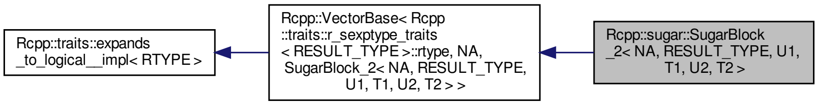
6.755 Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > Class Template Reference

`#include <SugarBlock_2.h>`

Inheritance diagram for Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >:



Collaboration diagram for Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >:



Public Types

- typedef RESULT_TYPE(∗ FunPtr) (U1, U2)

Public Member Functions

- SugarBlock_2 (FunPtr ptr_, const T1 &x_, const T2 &y_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const T1 & x
- const T2 & y

6.755.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2, typename T2>
class Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >

Definition at line 29 of file SugarBlock_2.h.

6.755.2 Member Typedef Documentation

6.755.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::FunPtr) (U1,
U2)
```

Definition at line 31 of file SugarBlock_2.h.

6.755.3 Constructor & Destructor Documentation

6.755.3.1 SugarBlock_2()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::SugarBlock_2 (
            FunPtr ptr_,
            const T1 & x_,
            const T2 & y_ )  [inline]
```

Definition at line 32 of file SugarBlock_2.h.

6.755.4 Member Function Documentation

6.755.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 37 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::ptr, Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::x, and Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::y.

6.755.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
R_xlen_t Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::size ( ) const  [inline]
```

Definition at line 40 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::x.

6.755.5 Member Data Documentation

6.755.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
FunPtr Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::ptr  [private]
```

Definition at line 43 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::operator[ ]().

6.755.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
const T1& Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::x  [private]
```

Definition at line 44 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::operator[ ](), and Rcpp::sugar::↩
SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::size().

6.755.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
const T2& Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::y  [private]
```

Definition at line 45 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::operator[ ]().

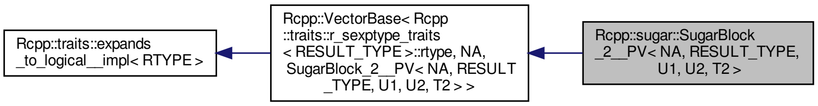
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

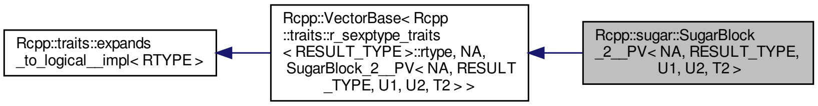
6.756 Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > Class Template Reference

```
#include <SugarBlock_2.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < RESULT_TYPE >::rtype, NA, SugarBlock_2__PV< NA, RESULT _TYPE, U1, U2, T2 > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < RESULT_TYPE >::rtype, NA, SugarBlock_2__PV< NA, RESULT _TYPE, U1, U2, T2 > >
Rcpp::sugar::SugarBlock _2__PV< NA, RESULT_TYPE, U1, U2, T2 >

Public Types

- typedef RESULT_TYPE(∗ FunPtr) (U1, U2)

Public Member Functions

- SugarBlock_2__PV (FunPtr ptr_, U1 u1, const T2 &y_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- U1 x
- const T2 & y

6.756.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename U2, typename T2>
class Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >

Definition at line 68 of file SugarBlock_2.h.

6.756.2 Member Typedef Documentation

6.756.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::FunPtr) (U1,
U2)
```

Definition at line 70 of file SugarBlock_2.h.

6.756.3 Constructor & Destructor Documentation

6.756.3.1 SugarBlock_2__PV()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::SugarBlock_2__PV (
             FunPtr ptr_,
             U1 u1,
             const T2 & y_ )  [inline]
```

Definition at line 71 of file SugarBlock_2.h.

6.756.4 Member Function Documentation

6.756.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 74 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::ptr, Rcpp::sugar::SugarBlock_2_↩_PV< NA, RESULT_TYPE, U1, U2, T2 >::x, and Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::y.

6.756.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
R_xlen_t Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::size ( ) const  [inline]
```

Definition at line 77 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::y.

6.756.5 Member Data Documentation

6.756.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
FunPtr Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::ptr  [private]
```

Definition at line 80 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[ ]().

6.756.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
U1 Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::x  [private]
```

Definition at line 81 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[ ]().

6.756.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
const T2& Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::y  [private]
```

Definition at line 82 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[ ](), and Rcpp::sugar↩::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::size().

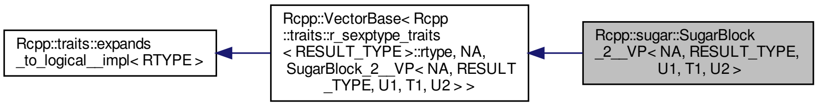
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

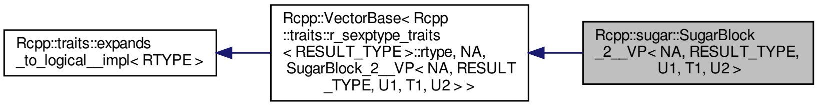
6.757 Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > Class Template Reference

```
#include <SugarBlock_2.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > >
Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >

Public Types

- typedef RESULT_TYPE(∗ FunPtr) (U1, U2)

Public Member Functions

- SugarBlock_2__VP (FunPtr ptr_, const T1 &x_, U2 u2)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const T1 & x
- U2 y

6.757.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2>
class Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >

Definition at line 50 of file SugarBlock_2.h.

6.757.2 Member Typedef Documentation

6.757.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::FunPtr) (U1,
U2)
```

Definition at line 52 of file SugarBlock_2.h.

6.757.3 Constructor & Destructor Documentation

6.757.3.1 SugarBlock_2__VP()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::SugarBlock_2__VP (
             FunPtr ptr_,
             const T1 & x_,
             U2 u2 )  [inline]
```

Definition at line 53 of file SugarBlock_2.h.

6.757.4 Member Function Documentation

6.757.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 56 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::ptr, Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::x, and Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::y.

6.757.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
R_xlen_t Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::size ( ) const  [inline]
```

Definition at line 59 of file SugarBlock_2.h.

References Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::x.

6.757.5 Member Data Documentation

6.757.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
FunPtr Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::ptr  [private]
```

Definition at line 62 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ]().

6.757.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
const T1& Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::x  [private]
```

Definition at line 63 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ](), and Rcpp::sugar↩ ::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::size().

6.757.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
U2 Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::y  [private]
```

Definition at line 64 of file SugarBlock_2.h.

Referenced by Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[ ]().

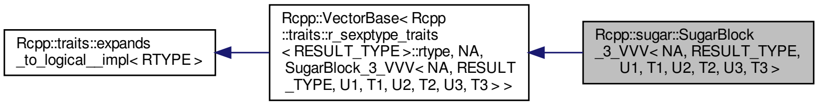
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_2.h

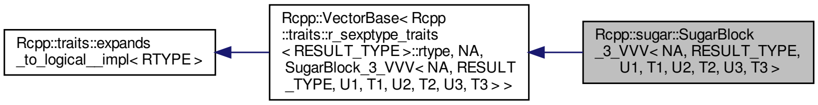
6.758 Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > Class Template Reference

```
#include <SugarBlock_3.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < RESULT_TYPE >::rtype, NA, SugarBlock_3_VVV< NA, RESULT _TYPE, U1, T1, U2, T2, U3, T3 > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < RESULT_TYPE >::rtype, NA, SugarBlock_3_VVV< NA, RESULT _TYPE, U1, T1, U2, T2, U3, T3 > >
Rcpp::sugar::SugarBlock _3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >

Public Types

- typedef RESULT_TYPE(∗ FunPtr) (U1, U2, U3)

Public Member Functions

- SugarBlock_3_VVV (FunPtr ptr_, const T1 &x_, const T2 &y_, const T3 &z_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const T1 & x
- const T2 & y
- const T2 & z

6.758.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2, typename T2, typename U3, typename T3> class Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >

Definition at line 34 of file SugarBlock_3.h.

6.758.2 Member Typedef Documentation

6.758.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >↩
::FunPtr) (U1, U2, U3)
```

Definition at line 39 of file SugarBlock_3.h.

6.758.3 Constructor & Destructor Documentation

6.758.3.1 SugarBlock_3_VVV()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::SugarBlock_3_VVV (
             FunPtr ptr_,
             const T1 & x_,
             const T2 & y_,
             const T3 & z_ )  [inline]
```

Definition at line 40 of file SugarBlock_3.h.

6.758.4 Member Function Documentation

6.758.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
RESULT_TYPE Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[]
(
            R_xlen_t i ) const  [inline]
```

Definition at line 44 of file SugarBlock_3.h.

References Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::ptr, Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::x, Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::y, and Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::z.

6.758.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
R_xlen_t Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::size ( ) const
[inline]
```

Definition at line 47 of file SugarBlock_3.h.

References Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::x.

6.758.5 Member Data Documentation

6.758.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
FunPtr Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::ptr  [private]
```

Definition at line 50 of file SugarBlock_3.h.

Referenced by Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[ ]().

6.758.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T1& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::x  [private]
```

Definition at line 51 of file SugarBlock_3.h.

Referenced by Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[ ](), and Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::size().

6.758.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T2& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::y  [private]
```

Definition at line 52 of file SugarBlock_3.h.

Referenced by Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[ ]().

6.758.5.4 z

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T2& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::z  [private]
```

Definition at line 53 of file SugarBlock_3.h.

Referenced by Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[ ]().

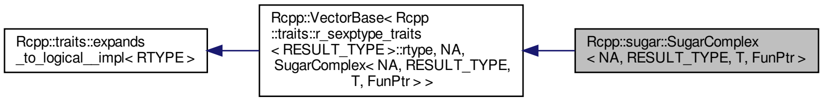
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock_3.h

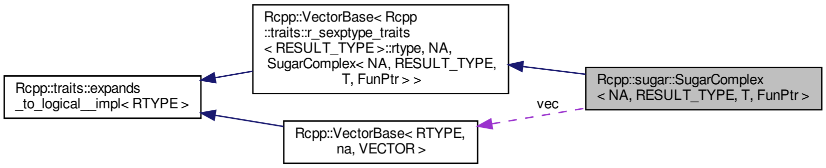
6.759 Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr > Class Template Reference

`#include <complex.h>`

Inheritance diagram for Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp
::traits::r_sexptype_traits
< RESULT_TYPE >::rtype, NA,
SugarComplex< NA, RESULT_TYPE,
T, FunPtr > >
Rcpp::sugar::SugarComplex
< NA, RESULT_TYPE, T, FunPtr >

Collaboration diagram for Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >:


Rcpp::VectorBase< Rcpp
::traits::r_sexptype_traits
< RESULT_TYPE >::rtype, NA,
SugarComplex< NA, RESULT_TYPE,
T, FunPtr > >
Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::sugar::SugarComplex
< NA, RESULT_TYPE, T, FunPtr >
vec
Rcpp::VectorBase< RTYPE,
na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< CPLXSXP, NA, T > VEC_TYPE

Public Member Functions

- SugarComplex (FunPtr ptr_, const VEC_TYPE &vec_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec

6.759.1 Detailed Description

**template<bool NA, typename RESULT_TYPE, typename T, typename FunPtr>
class Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >**

Definition at line 30 of file complex.h.

6.759.2 Member Typedef Documentation

6.759.2.1 VEC_TYPE

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >↩
::VEC_TYPE
```

Definition at line 37 of file complex.h.

6.759.3 Constructor & Destructor Documentation

6.759.3.1 SugarComplex()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::SugarComplex (
             FunPtr ptr_,
             const VEC_TYPE & vec_ )  [inline]
```

Definition at line 39 of file complex.h.

6.759.4 Member Function Documentation

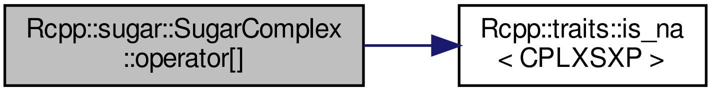
6.759.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 41 of file complex.h.

References Rcpp::traits::is_na< 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:



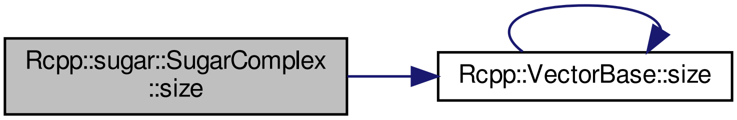
6.759.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size ( ) const [inline]
```

Definition at line 47 of file complex.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::vec.

Here is the call graph for this function:



6.759.5 Member Data Documentation

6.759.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
FunPtr Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::ptr [private]
```

Definition at line 50 of file complex.h.

Referenced by Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[ ]().

6.759.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::vec [private]
```

Definition at line 51 of file complex.h.

Referenced by Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[ ](), and Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/complex.h

6.760 Rcpp::sugar::SugarIterator< T > Class Template Reference

```
#include <iterator.h>
```

Public Types

- typedef R_xlen_t difference_type
- typedef Rcpp::traits::storage_type< Rcpp::traits::r_sexptype_traits< T >::rtype >::type STORAGE_TYPE
- typedef STORAGE_TYPE reference
- typedef STORAGE_TYPE ∗ pointer
- typedef std::random_access_iterator_tag iterator_category
- typedef SugarIterator iterator

Public Member Functions

- SugarIterator (const T &ref_)
- SugarIterator (const T &ref_, R_xlen_t index_)
- SugarIterator (const SugarIterator &other)
- iterator & operator++ ()
- iterator operator++ (int)
- iterator & operator-- ()
- iterator operator-- (int)
- iterator operator+ (difference_type n) const
- iterator operator- (difference_type n) const
- iterator & operator+= (difference_type n)
- iterator & operator-= (difference_type n)
- reference operator[ ] (R_xlen_t i)
- reference operator∗ ()
- pointer operator-> ()
- bool operator== (const iterator &y) const
- bool operator!= (const iterator &y) const
- bool operator< (const iterator &other) const
- bool operator> (const iterator &other) const
- bool operator<= (const iterator &other) const
- bool operator>= (const iterator &other) const
- difference_type operator- (const iterator &other) const

Private Attributes

- const T & ref
- R_xlen_t index

6.760.1 Detailed Description

template<typename T>
class Rcpp::sugar::SugarIterator< T >

Definition at line 30 of file iterator.h.

6.760.2 Member Typedef Documentation

6.760.2.1 difference_type

```
template<typename T >
typedef R_xlen_t Rcpp::sugar::SugarIterator< T >::difference_type
```

Definition at line 33 of file iterator.h.

6.760.2.2 iterator

```
template<typename T >
typedef SugarIterator Rcpp::sugar::SugarIterator< T >::iterator
```

Definition at line 38 of file iterator.h.

6.760.2.3 iterator_category

```
template<typename T >
typedef std::random_access_iterator_tag Rcpp::sugar::SugarIterator< T >::iterator_category
```

Definition at line 37 of file iterator.h.

6.760.2.4 pointer

```
template<typename T >
typedef STORAGE_TYPE* Rcpp::sugar::SugarIterator< T >::pointer
```

Definition at line 36 of file iterator.h.

6.760.2.5 reference

```
template<typename T >
typedef STORAGE_TYPE Rcpp::sugar::SugarIterator< T >::reference
```

Definition at line 35 of file iterator.h.

6.760.2.6 STORAGE_TYPE

```
template<typename T >
typedef Rcpp::traits::storage_type< Rcpp::traits::r_sexptype_traits<T>::rtype >::type Rcpp::sugar::SugarIterator
T >::STORAGE_TYPE
```

Definition at line 34 of file iterator.h.

6.760.3 Constructor & Destructor Documentation

6.760.3.1 SugarIterator() [1/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
             const T & ref_ )  [inline]
```

Definition at line 40 of file iterator.h.

6.760.3.2 SugarIterator() [2/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
             const T & ref_,
             R_xlen_t index_ )  [inline]
```

Definition at line 41 of file iterator.h.

6.760.3.3 SugarIterator() [3/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
             const SugarIterator< T > & other )  [inline]
```

Definition at line 42 of file iterator.h.

6.760.4 Member Function Documentation

6.760.4.1 operator"!=()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator!= (
             const iterator & y ) const  [inline]
```

Definition at line 84 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.2 operator∗()

```
template<typename T >
reference Rcpp::sugar::SugarIterator< T >::operator∗ ( )  [inline]
```

Definition at line 74 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

6.760.4.3 operator+()

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator+ (
             difference_type n ) const  [inline]
```

Definition at line 56 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

6.760.4.4 operator++() [1/2]

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator++ ( )  [inline]
```

Definition at line 44 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.5 operator++() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator++ (
             int  )  [inline]
```

Definition at line 45 of file iterator.h.

6.760.4.6 operator+=()

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator+= (
             difference_type n )  [inline]
```

Definition at line 62 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.7 operator-() [1/2]

```
template<typename T >
difference_type Rcpp::sugar::SugarIterator< T >::operator- (
             const iterator & other ) const  [inline]
```

Definition at line 100 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.8 operator-() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator- (
             difference_type n ) const  [inline]
```

Definition at line 59 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

6.760.4.9 operator--() [1/2]

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator-- ( )  [inline]
```

Definition at line 50 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.10 operator--() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator-- (
             int  )  [inline]
```

Definition at line 51 of file iterator.h.

6.760.4.11 operator-=()

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator-= (
             difference_type n )  [inline]
```

Definition at line 66 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.12 operator->()

```
template<typename T >
pointer Rcpp::sugar::SugarIterator< T >::operator-> ( )  [inline]
```

Definition at line 77 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

6.760.4.13 operator<()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator< (
             const iterator & other ) const  [inline]
```

Definition at line 87 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.14 operator<=()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator<= (
             const iterator & other ) const  [inline]
```

Definition at line 93 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.15 operator==()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator== (
             const iterator & y ) const  [inline]
```

Definition at line 81 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.16 operator>()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator> (
             const iterator & other ) const  [inline]
```

Definition at line 90 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.17 operator>=()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator>= (
             const iterator & other ) const  [inline]
```

Definition at line 96 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

6.760.4.18 operator[]()

```
template<typename T >
reference Rcpp::sugar::SugarIterator< T >::operator[ ] (
             R_xlen_t i )  [inline]
```

Definition at line 70 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

6.760.5 Member Data Documentation

6.760.5.1 index

```
template<typename T >
R_xlen_t Rcpp::sugar::SugarIterator< T >::index  [private]
```

Definition at line 107 of file iterator.h.

Referenced by Rcpp::sugar::SugarIterator< T >::operator!=(), Rcpp::sugar::SugarIterator< T >::operator∗(), Rcpp::sugar::SugarIterator< T >::operator+(), Rcpp::sugar::SugarIterator< T >::operator++(), Rcpp::sugar::SugarIterator< T >::operator+=(), Rcpp::sugar::SugarIterator< T >::operator-(), Rcpp::sugar::SugarIterator< T >::operator--(), Rcpp::sugar::SugarIterator< T >::operator-=(), Rcpp::sugar::SugarIterator< T >::operator->(), Rcpp::sugar::SugarIterator< T >::operator<(), Rcpp::sugar::SugarIterator< T >::operator<=(), Rcpp::sugar::SugarIterator< T >::operator==(), Rcpp::sugar::SugarIterator< T >::operator>(), Rcpp::sugar::SugarIterator< T >::operator>=(), and Rcpp::sugar::SugarIterator< T >::operator[ ]().

6.760.5.2 ref

```
template<typename T >
const T& Rcpp::sugar::SugarIterator< T >::ref [private]
```

Definition at line 106 of file iterator.h.

Referenced by Rcpp::sugar::SugarIterator< T >::operator∗(), Rcpp::sugar::SugarIterator< T >::operator+(), Rcpp::sugar::SugarIterator< T >::operator-(), Rcpp::sugar::SugarIterator< T >::operator->(), and Rcpp::sugar::SugarIterator< T >::operator[ ]().

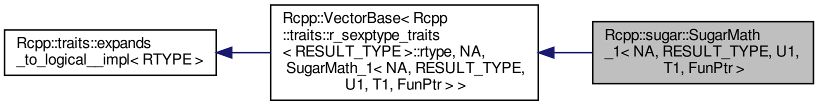
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

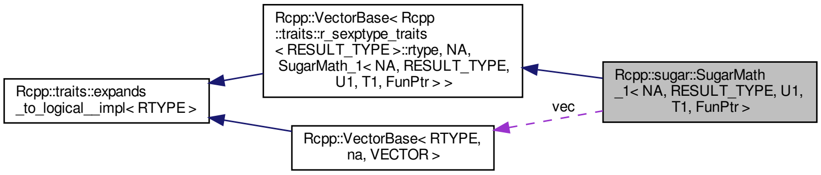
6.761 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >

Collaboration diagram for Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >:


Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >
vec
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype,NA, T1 > VEC_TYPE

Public Member Functions

- SugarMath_1 (FunPtr ptr_, const VEC_TYPE &vec_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec

6.761.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >

Definition at line 29 of file SugarMath.h.

6.761.2 Member Typedef Documentation

6.761.2.1 VEC_TYPE

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits<RESULT_TYPE>::rtype ,NA,T1> Rcpp::sugar::SugarMath_1<
NA, RESULT_TYPE, U1, T1, FunPtr >::VEC_TYPE
```

Definition at line 36 of file SugarMath.h.

6.761.3 Constructor & Destructor Documentation

6.761.3.1 SugarMath_1()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::SugarMath_1 (
             FunPtr ptr_,
             const VEC_TYPE & vec_ )  [inline]
```

Definition at line 38 of file SugarMath.h.

6.761.4 Member Function Documentation

6.761.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 40 of file SugarMath.h.

References Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

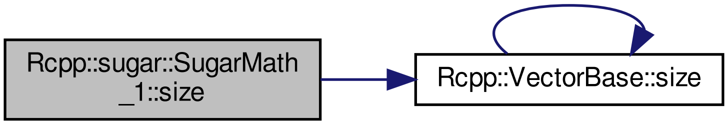
6.761.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::size ( ) const  [inline]
```

Definition at line 45 of file SugarMath.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:



6.761.5 Member Data Documentation

6.761.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr  [private]
```

Definition at line 48 of file SugarMath.h.

Referenced by Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[ ](), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[ ](), and Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[ ]().

6.761.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec  [private]
```

Definition at line 49 of file SugarMath.h.

Referenced by Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[ ](), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[ ](), Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[ ](), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::size(), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size(), and Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size().

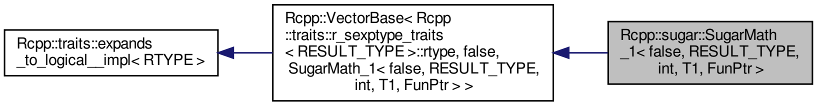
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

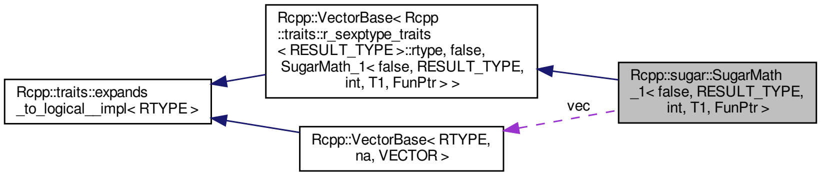
6.762 Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >:



Collaboration diagram for Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >:


Rcpp::VectorBase< Rcpp ::traits::r_sexptype_traits < RESULT_TYPE >::rtype, false, SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
vec
Rcpp::sugar::SugarMath _1< false, RESULT_TYPE, int, T1, FunPtr >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< INTSXP,false, T1 > VEC_TYPE

Public Member Functions

- SugarMath_1 (FunPtr ptr_, const VEC_TYPE &vec_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec

6.762.1 Detailed Description

template<typename RESULT_TYPE, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >

Definition at line 76 of file SugarMath.h.

6.762.2 Member Typedef Documentation

6.762.2.1 VEC_TYPE

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< INTSXP ,false,T1> Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int,
T1, FunPtr >::VEC_TYPE
```

Definition at line 83 of file SugarMath.h.

6.762.3 Constructor & Destructor Documentation

6.762.3.1 SugarMath_1()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::SugarMath_1 (
            FunPtr ptr_,
            const VEC_TYPE & vec_ ) [inline]
```

Definition at line 84 of file SugarMath.h.

6.762.4 Member Function Documentation

6.762.4.1 operator[]()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[] (
            R_xlen_t i ) const [inline]
```

Definition at line 86 of file SugarMath.h.

References Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

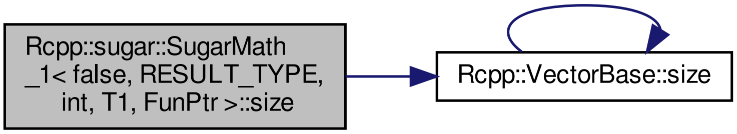
6.762.4.2 size()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size ( ) const [inline]
```

Definition at line 89 of file SugarMath.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:



6.762.5 Member Data Documentation

6.762.5.1 ptr

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::ptr  [private]
```

Definition at line 92 of file SugarMath.h.

6.762.5.2 vec

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::vec  [private]
```

Definition at line 93 of file SugarMath.h.

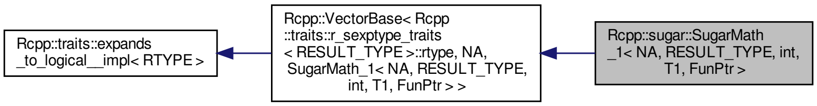
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

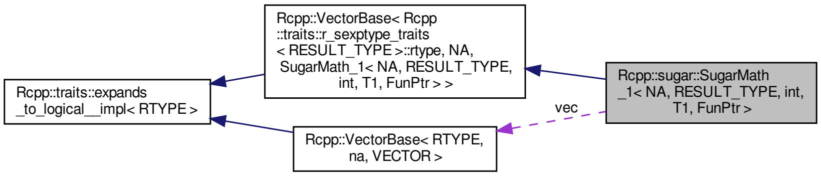
6.763 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< Rcpp
::traits::r_sexptype_traits
< RESULT_TYPE >::rtype, NA,
SugarMath_1< NA, RESULT_TYPE,
int, T1, FunPtr > >
Rcpp::sugar::SugarMath
_1< NA, RESULT_TYPE, int,
T1, FunPtr >

Collaboration diagram for Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >:



Public Types

- typedef Rcpp::VectorBase< INTSXP,NA, T1 > VEC_TYPE

Public Member Functions

- SugarMath_1 (FunPtr ptr_, const VEC_TYPE &vec_)
- RESULT_TYPE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- FunPtr ptr
- const VEC_TYPE & vec

6.763.1 Detailed Description

template<bool NA, typename RESULT_TYPE, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >

Definition at line 53 of file SugarMath.h.

6.763.2 Member Typedef Documentation

6.763.2.1 VEC_TYPE

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< INTSXP ,NA,T1> Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1,
FunPtr >::VEC_TYPE
```

Definition at line 59 of file SugarMath.h.

6.763.3 Constructor & Destructor Documentation

6.763.3.1 SugarMath_1()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::SugarMath_1 (
            FunPtr ptr_,
            const VEC_TYPE & vec_ )  [inline]
```

Definition at line 61 of file SugarMath.h.

6.763.4 Member Function Documentation

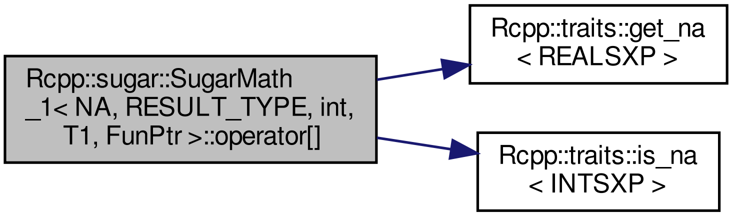
6.763.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 63 of file SugarMath.h.

References Rcpp::traits::get_na< REALSXP >(), Rcpp::traits::is_na< INTSXP >(), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:


Rcpp::sugar::SugarMath _1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[]
Rcpp::traits::get_na < REALSXP >
Rcpp::traits::is_na < INTSXP >

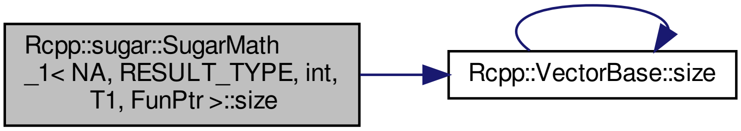
6.763.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size ( ) const  [inline]
```

Definition at line 68 of file SugarMath.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:



6.763.5 Member Data Documentation

6.763.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::ptr  [private]
```

Definition at line 71 of file SugarMath.h.

6.763.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::vec  [private]
```

Definition at line 72 of file SugarMath.h.

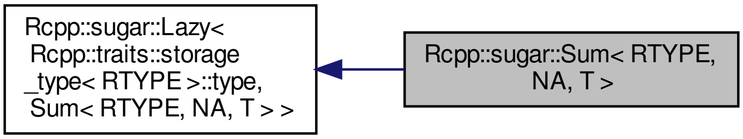
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarMath.h

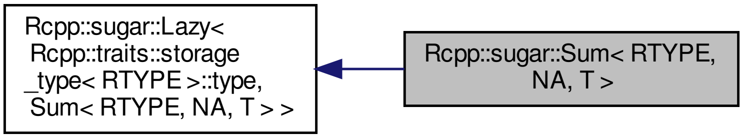
6.764 Rcpp::sugar::Sum< RTYPE, NA, T > Class Template Reference

```
#include <sum.h>
```

Inheritance diagram for Rcpp::sugar::Sum< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sum< RTYPE, NA, T > >
Rcpp::sugar::Sum< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Sum< RTYPE, NA, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sum< RTYPE, NA, T > >
Rcpp::sugar::Sum< RTYPE, NA, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type VEC_EXT

Public Member Functions

- Sum (const VEC_TYPE &object_)
- STORAGE get () const

Private Attributes

- const VEC_EXT & object

6.764.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sum< RTYPE, NA, T >

Definition at line 29 of file sum.h.

6.764.2 Member Typedef Documentation

6.764.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sum< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file sum.h.

6.764.2.2 VEC_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sum< RTYPE, NA, T >::VEC_EXT
```

Definition at line 33 of file sum.h.

6.764.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sum< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file sum.h.

6.764.3 Constructor & Destructor Documentation

6.764.3.1 Sum()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sum< RTYPE, NA, T >::Sum (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 35 of file sum.h.

6.764.4 Member Function Documentation

6.764.4.1 get()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Sum< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 37 of file sum.h.

6.764.5 Member Data Documentation

6.764.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Sum< RTYPE, NA, T >::object  [private]
```

Definition at line 50 of file sum.h.

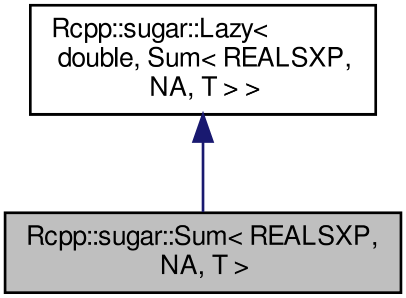
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

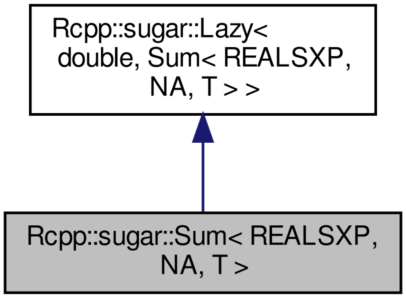
6.765 Rcpp::sugar::Sum< REALSXP, NA, T > Class Template Reference

`#include <sum.h>`

Inheritance diagram for Rcpp::sugar::Sum< REALSXP, NA, T >:


Rcpp::sugar::Lazy< double, Sum< REALSXP, NA, T > >
Rcpp::sugar::Sum< REALSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Sum< REALSXP, NA, T >:


Rcpp::sugar::Lazy< double, Sum< REALSXP, NA, T > >
Rcpp::sugar::Sum< REALSXP, NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type VEC_EXT

Public Member Functions

- Sum (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_EXT & object

6.765.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Sum< REALSXP, NA, T >

Definition at line 54 of file sum.h.

6.765.2 Member Typedef Documentation

6.765.2.1 VEC_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor< REALSXP, NA, T>::type Rcpp::sugar::Sum< REALSXP, NA, T >↩
::VEC_EXT
```

Definition at line 57 of file sum.h.

6.765.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Sum< REALSXP, NA, T >::VEC_TYPE
```

Definition at line 56 of file sum.h.

6.765.3 Constructor & Destructor Documentation

6.765.3.1 Sum()

```
template<bool NA, typename T >
Rcpp::sugar::Sum< REALSXP, NA, T >::Sum (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 59 of file sum.h.

6.765.4 Member Function Documentation

6.765.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Sum< REALSXP, NA, T >::get ( ) const  [inline]
```

Definition at line 61 of file sum.h.

6.765.5 Member Data Documentation

6.765.5.1 object

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Sum< REALSXP, NA, T >::object  [private]
```

Definition at line 70 of file sum.h.

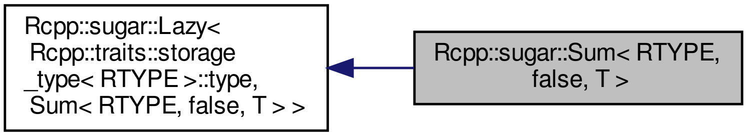
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

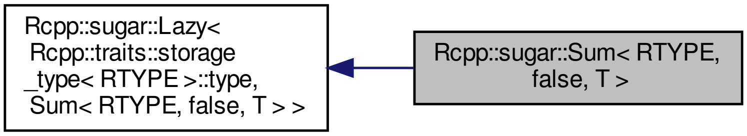
6.766 Rcpp::sugar::Sum< RTYPE, false, T > Class Template Reference

```
#include <sum.h>
```

Inheritance diagram for Rcpp::sugar::Sum< RTYPE, false, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sum< RTYPE, false, T > >
Rcpp::sugar::Sum< RTYPE, false, T >

Collaboration diagram for Rcpp::sugar::Sum< RTYPE, false, T >:


Rcpp::sugar::Lazy< Rcpp::traits::storage _type< RTYPE >::type, Sum< RTYPE, false, T > >
Rcpp::sugar::Sum< RTYPE, false, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type VEC_EXT

Public Member Functions

- Sum (const VEC_TYPE &object_)
- STORAGE get () const

Private Attributes

- const VEC_EXT & object

6.766.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Sum< RTYPE, false, T >

Definition at line 75 of file sum.h.

6.766.2 Member Typedef Documentation

6.766.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sum< RTYPE, false, T >::STORAGE
```

Definition at line 78 of file sum.h.

6.766.2.2 VEC_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Sum< RTYPE, false, T >↩
::VEC_EXT
```

Definition at line 79 of file sum.h.

6.766.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Sum< RTYPE, false, T >::VEC_TYPE
```

Definition at line 77 of file sum.h.

6.766.3 Constructor & Destructor Documentation

6.766.3.1 Sum()

```
template<int RTYPE, typename T >
Rcpp::sugar::Sum< RTYPE, false, T >::Sum (
            const VEC_TYPE & object_ )  [inline]
```

Definition at line 81 of file sum.h.

6.766.4 Member Function Documentation

6.766.4.1 get()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Sum< RTYPE, false, T >::get ( ) const  [inline]
```

Definition at line 83 of file sum.h.

6.766.5 Member Data Documentation

6.766.5.1 object

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Sum< RTYPE, false, T >::object  [private]
```

Definition at line 92 of file sum.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

6.767 Rcpp::SuspendRNGSynchronizationScope Class Reference

```
#include <RNGScope.h>
```

Public Member Functions

- SuspendRNGSynchronizationScope ()
- ∼SuspendRNGSynchronizationScope ()

6.767.1 Detailed Description

Definition at line 33 of file RNGScope.h.

6.767.2 Constructor & Destructor Documentation

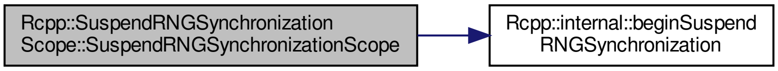
6.767.2.1 SuspendRNGSynchronizationScope()

```
Rcpp::SuspendRNGSynchronizationScope::SuspendRNGSynchronizationScope ( )  [inline]
```

Definition at line 35 of file RNGScope.h.

References Rcpp::internal::beginSuspendRNGSynchronization().

Here is the call graph for this function:


Rcpp::SuspendRNGSynchronization
Scope::SuspendRNGSynchronizationScope
Rcpp::internal::beginSuspend
RNGSynchronization

6.767.2.2 ∼SuspendRNGSynchronizationScope()

```
Rcpp::SuspendRNGSynchronizationScope::∼SuspendRNGSynchronizationScope ( )  [inline]
```

Definition at line 36 of file RNGScope.h.

References Rcpp::internal::endSuspendRNGSynchronization().

Here is the call graph for this function:

Rcpp::SuspendRNGSynchronization
Scope::~SuspendRNGSynchronizationScope
Rcpp::internal::endSuspend
RNGSynchronization

The documentation for this class was generated from the following file:

- inst/include/Rcpp/RNGScope.h

6.768 Rcpp::sugar::Table< RTYPE, TABLE_T > Class Template Reference

```
#include <table.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Table (const TABLE_T &table)
- operator IntegerVector () const

Private Types

- typedef RCPP_UNORDERED_MAP< STORAGE, int > HASH
- typedef CountInserter< HASH, STORAGE > Inserter
- typedef std::map< STORAGE, int, internal::NAComparator< STORAGE > > SORTED_MAP

Private Attributes

- HASH hash
- SORTED_MAP map

6.768.1 Detailed Description

template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::Table< RTYPE, TABLE_T >

Definition at line 89 of file table.h.

6.768.2 Member Typedef Documentation

6.768.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef RCPP_UNORDERED_MAP<STORAGE, int> Rcpp::sugar::Table< RTYPE, TABLE_T >::HASH  [private]
```

Definition at line 112 of file table.h.

6.768.2.2 Inserter

```
template<int RTYPE, typename TABLE_T >
typedef CountInserter<HASH,STORAGE> Rcpp::sugar::Table< RTYPE, TABLE_T >::Inserter  [private]
```

Definition at line 113 of file table.h.

6.768.2.3 SORTED_MAP

```
template<int RTYPE, typename TABLE_T >
typedef std::map<STORAGE, int, internal::NAComparator<STORAGE> > Rcpp::sugar::Table< RTYPE,
TABLE_T >::SORTED_MAP  [private]
```

Definition at line 116 of file table.h.

6.768.2.4 STORAGE

```
template<int RTYPE, typename TABLE_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Table< RTYPE, TABLE_T >::STORAGE
```

Definition at line 91 of file table.h.

6.768.3 Constructor & Destructor Documentation

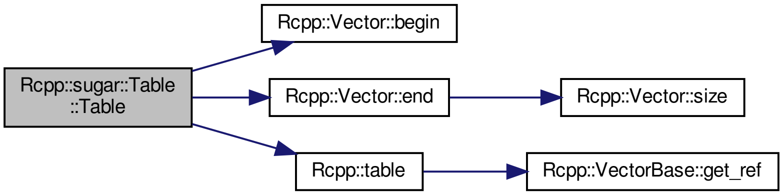
6.768.3.1 Table()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::Table< RTYPE, TABLE_T >::Table (
            const TABLE_T & table )  [inline]
```

Definition at line 93 of file table.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::sugar::Table< RTYPE, TABLE_T >::hash, Rcpp::sugar::Table< RTYPE, TABLE_T >::map, and Rcpp::table().

Here is the call graph for this function:



6.768.4 Member Function Documentation

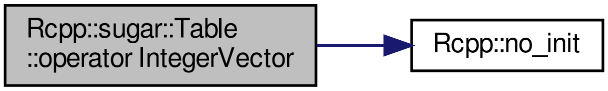
6.768.4.1 operator IntegerVector()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector ( ) const  [inline]
```

Definition at line 101 of file table.h.

References Rcpp::sugar::Table< RTYPE, TABLE_T >::map, and Rcpp::no_init().

Here is the call graph for this function:



6.768.5 Member Data Documentation

6.768.5.1 hash

```
template<int RTYPE, typename TABLE_T >
HASH Rcpp::sugar::Table< RTYPE, TABLE_T >::hash  [private]
```

Definition at line 114 of file table.h.

Referenced by Rcpp::sugar::Table< RTYPE, TABLE_T >::Table().

6.768.5.2 map

```
template<int RTYPE, typename TABLE_T >
SORTED_MAP Rcpp::sugar::Table< RTYPE, TABLE_T >::map  [private]
```

Definition at line 117 of file table.h.

Referenced by Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector(), and Rcpp::sugar::Table< RTYPE, TABLE_T >::Table().

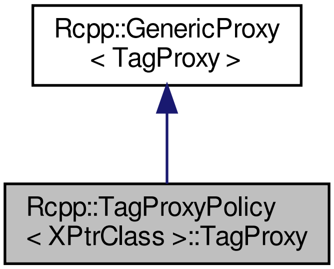
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/table.h

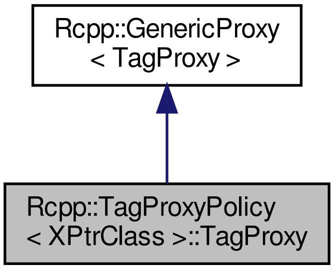
6.769 Rcpp::TagProxyPolicy< XPtrClass >::TagProxy Class Reference

```
#include <TagProxy.h>
```

Inheritance diagram for Rcpp::TagProxyPolicy< XPtrClass >::TagProxy:



Collaboration diagram for Rcpp::TagProxyPolicy< XPtrClass >::TagProxy:



Public Member Functions

- TagProxy (XPtrClass &xp_)
- template< typename U >
  TagProxy & operator= (const U &u)
- template< typename U >
  operator U () const
- operator SEXP () const
- template< typename T >
  TagProxyPolicy< CLASS >::TagProxy & operator= (const T &rhs)

Private Member Functions

- SEXP get () const
- void set (SEXP x)

Private Attributes

- XPtrClass & xp

6.769.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::TagProxy

Definition at line 27 of file TagProxy.h.

6.769.2 Constructor & Destructor Documentation

6.769.2.1 TagProxy()

```
template<typename XPtrClass >
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::TagProxy (
            XPtrClass & xp_ )  [inline]
```

Definition at line 29 of file TagProxy.h.

6.769.3 Member Function Documentation

6.769.3.1 get()

```
template<typename XPtrClass >
SEXP Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::get ( ) const  [inline], [private]
```

Definition at line 43 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp.

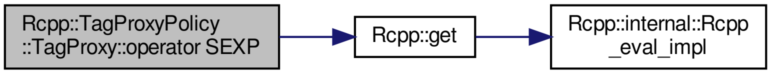
6.769.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::TagProxyPolicy< CLASS >::TagProxy::operator SEXP
```

Definition at line 111 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:


Rcpp::TagProxyPolicy ::TagProxy::operator SEXP
Rcpp::get
Rcpp::internal::Rcpp _eval_impl

6.769.3.3 operator U()

```
template<typename XPtrClass >
template<typename U >
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator U ( ) const
```

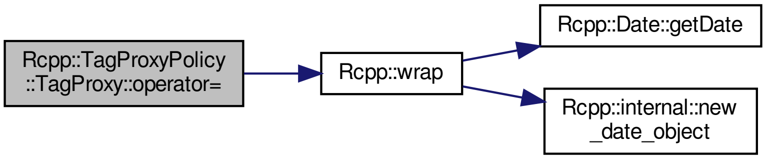
6.769.3.4 operator=() [1/2]

```
template<typename XPtrClass >
template<typename T >
TagProxyPolicy<CLASS>::TagProxy& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator= (
            const T & rhs )
```

Definition at line 99 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:


Rcpp::TagProxyPolicy ::TagProxy::operator=
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.769.3.5 operator=() [2/2]

```
template<typename XPtrClass >
template<typename U >
TagProxy& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator= (
             const U & u )
```

6.769.3.6 set()

```
template<typename XPtrClass >
void Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::set (
             SEXP x )  [inline], [private]
```

Definition at line 47 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp.

6.769.4 Member Data Documentation

6.769.4.1 xp

```
template<typename XPtrClass >
XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp  [private]
```

Definition at line 41 of file TagProxy.h.

Referenced by Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::get(), and Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::set().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/TagProxy.h
- inst/include/Rcpp/api/meat/proxy.h

6.770 Rcpp::TagProxyPolicy< XPtrClass > Class Template Reference

```
#include <TagProxy.h>
```

Classes

- class const_TagProxy
- class TagProxy

Public Member Functions

- TagProxy tag ()
- const_TagProxy tag () const

6.770.1 Detailed Description

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >

Definition at line 24 of file TagProxy.h.

6.770.2 Member Function Documentation

6.770.2.1 tag() [1/2]

```
template<typename XPtrClass >
TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( )  [inline]
```

Definition at line 73 of file TagProxy.h.

6.770.2.2 tag() [2/2]

```
template<typename XPtrClass >
const_TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( ) const  [inline]
```

Definition at line 77 of file TagProxy.h.

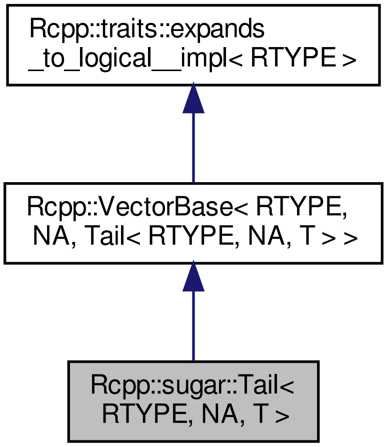
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/TagProxy.h

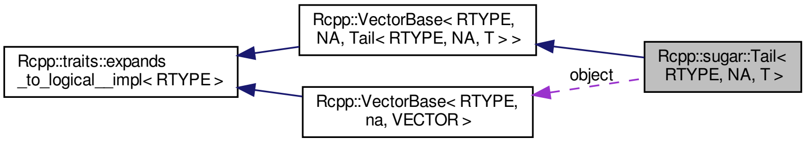
6.771 Rcpp::sugar::Tail< RTYPE, NA, T > Class Template Reference

```
#include <tail.h>
```

Inheritance diagram for Rcpp::sugar::Tail< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Tail< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Tail (const VEC_TYPE &object_, R_xlen_t n_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & object
- R_xlen_t start
- R_xlen_t n

6.771.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Tail< RTYPE, NA, T >

Definition at line 29 of file tail.h.

6.771.2 Member Typedef Documentation

6.771.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Tail< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file tail.h.

6.771.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Tail< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file tail.h.

6.771.3 Constructor & Destructor Documentation

6.771.3.1 Tail()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Tail< RTYPE, NA, T >::Tail (
             const VEC_TYPE & object_,
             R_xlen_t n_ )  [inline]
```

Definition at line 34 of file tail.h.

References Rcpp::sugar::Tail< RTYPE, NA, T >::n, and Rcpp::sugar::Tail< RTYPE, NA, T >::start.

6.771.4 Member Function Documentation

6.771.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Tail< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 43 of file tail.h.

References Rcpp::sugar::Tail< RTYPE, NA, T >::start.

6.771.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 46 of file tail.h.

References Rcpp::sugar::Tail< RTYPE, NA, T >::n.

6.771.5 Member Data Documentation

6.771.5.1 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::n  [private]
```

Definition at line 50 of file tail.h.

Referenced by Rcpp::sugar::Tail< RTYPE, NA, T >::size(), and Rcpp::sugar::Tail< RTYPE, NA, T >::Tail().

6.771.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Tail< RTYPE, NA, T >::object  [private]
```

Definition at line 49 of file tail.h.

6.771.5.3 start

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::start  [private]
```

Definition at line 50 of file tail.h.

Referenced by Rcpp::sugar::Tail< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Tail< RTYPE, NA, T >::Tail().

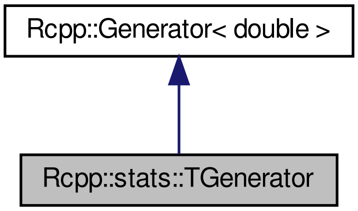
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/tail.h

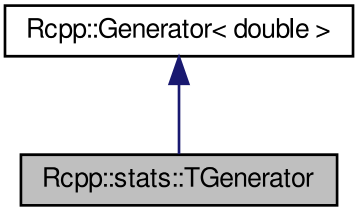
6.772 Rcpp::stats::TGenerator Class Reference

```
#include <rt.h>
```

Inheritance diagram for Rcpp::stats::TGenerator:



Collaboration diagram for Rcpp::stats::TGenerator:



Public Member Functions

- TGenerator (double df_)
- double operator() () const

Private Attributes

- double df
- double df_2

Additional Inherited Members

6.772.1 Detailed Description

Definition at line 28 of file rt.h.

6.772.2 Constructor & Destructor Documentation

6.772.2.1 TGenerator()

```
Rcpp::stats::TGenerator::TGenerator (
             double df_ )  [inline]
```

Definition at line 31 of file rt.h.

6.772.3 Member Function Documentation

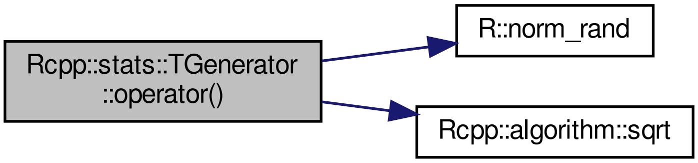
6.772.3.1 operator()()

```
double Rcpp::stats::TGenerator::operator() ( ) const  [inline]
```

Definition at line 33 of file rt.h.

References df, df_2, R::norm_rand(), and Rcpp::algorithm::sqrt().

Here is the call graph for this function:



6.772.4 Member Data Documentation

6.772.4.1 df

```
double Rcpp::stats::TGenerator::df  [private]
```

Definition at line 45 of file rt.h.

Referenced by operator()().

6.772.4.2 df_2

```
double Rcpp::stats::TGenerator::df_2  [private]
```

Definition at line 45 of file rt.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rt.h

6.773 Rcpp::Timer Class Reference

```
#include <Timer.h>
```

Public Member Functions

- Timer ()
- Timer (nanotime_t start_time_)
- void step (const std::string &name)
- operator SEXP () const
- nanotime_t now () const
- nanotime_t origin () const

Static Public Member Functions

- static std::vector< Timer > get_timers (int n)

Private Types

- typedef std::pair< std::string, nanotime_t > Step
- typedef std::vector< Step > Steps

Private Attributes

- Steps data
- const nanotime_t start_time

6.773.1 Detailed Description

Definition at line 103 of file Timer.h.

6.773.2 Member Typedef Documentation

6.773.2.1 Step

```
typedef std::pair<std::string,nanotime_t> Rcpp::Timer::Step  [private]
```

Definition at line 137 of file Timer.h.

6.773.2.2 Steps

```
typedef std::vector<Step> Rcpp::Timer::Steps  [private]
```

Definition at line 138 of file Timer.h.

6.773.3 Constructor & Destructor Documentation

6.773.3.1 Timer() [1/2]

```
Rcpp::Timer::Timer ( )  [inline]
```

Definition at line 105 of file Timer.h.

Referenced by get_timers().

6.773.3.2 Timer() [2/2]

```
Rcpp::Timer::Timer (
             nanotime_t start_time_ )  [inline]
```

Definition at line 106 of file Timer.h.

6.773.4 Member Function Documentation

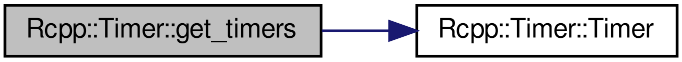
6.773.4.1 get_timers()

```
static std::vector<Timer> Rcpp::Timer::get_timers (
             int n )  [inline], [static]
```

Definition at line 124 of file Timer.h.

References Timer().

Here is the call graph for this function:



6.773.4.2 now()

```
nanotime_t Rcpp::Timer::now ( ) const  [inline]
```

Definition at line 128 of file Timer.h.

Referenced by step().

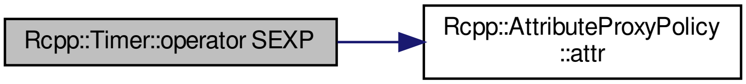
6.773.4.3 operator SEXP()

```
Rcpp::Timer::operator SEXP ( ) const  [inline]
```

Definition at line 112 of file Timer.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), data, and start_time.

Here is the call graph for this function:



6.773.4.4 origin()

```
nanotime_t Rcpp::Timer::origin ( ) const  [inline]
```

Definition at line 132 of file Timer.h.

References start_time.

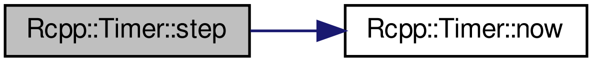
6.773.4.5 step()

```
void Rcpp::Timer::step (
             const std::string & name )  [inline]
```

Definition at line 108 of file Timer.h.

References data, and now().

Here is the call graph for this function:



6.773.5 Member Data Documentation

6.773.5.1 data

```
Steps Rcpp::Timer::data  [private]
```

Definition at line 140 of file Timer.h.

Referenced by operator SEXP(), and step().

6.773.5.2 start_time

```
const nanotime_t Rcpp::Timer::start_time  [private]
```

Definition at line 141 of file Timer.h.

Referenced by operator SEXP(), and origin().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Benchmark/Timer.h

6.774 Timer Class Reference

```
#include <Timer.h>
```

Public Member Functions

- Timer ()
- void Start ()
- void Stop ()
- void Reset ()
- double ElapsedTime ()
- double CumulativeTime ()

Private Member Functions

- double getFractionalSeconds (void)

Private Attributes

- Function sys_time
- double start_t
- double end_t
- double elapsed
- double cumul

6.774.1 Detailed Description

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 31 of file Timer.h.

6.774.2 Constructor & Destructor Documentation

6.774.2.1 Timer()

```
Timer::Timer ( )  [inline]
```

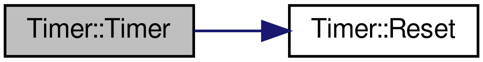
Examples

SugarPerformance/Timer.h.

Definition at line 33 of file Timer.h.

References Reset().

Here is the call graph for this function:



6.774.3 Member Function Documentation

6.774.3.1 CumulativeTime()

```
double Timer::CumulativeTime ( )  [inline]
```

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 42 of file Timer.h.

References cumul.

Referenced by main().

6.774.3.2 ElapsedTime()

```
double Timer::ElapsedTime ( )  [inline]
```

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 41 of file Timer.h.

References elapsed.

Referenced by main().

6.774.3.3 getFractionalSeconds()

```
double Timer::getFractionalSeconds (
            void  )  [inline], [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 49 of file Timer.h.

References sys_time.

Referenced by Start(), and Stop().

6.774.3.4 Reset()

```
void Timer::Reset ( )  [inline]
```

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 40 of file Timer.h.

References cumul, elapsed, end_t, and start_t.

Referenced by main(), and Timer().

6.774.3.5 Start()

```
void Timer::Start ( )  [inline]
```

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 34 of file Timer.h.

References getFractionalSeconds(), and start_t.

Referenced by main().

Here is the call graph for this function:

Timer::Start
Timer::getFractionalSeconds

6.774.3.6 Stop()

```
void Timer::Stop ( )  [inline]
```

Examples

SugarPerformance/Timer.h, and SugarPerformance/Timertest.cpp.

Definition at line 35 of file Timer.h.

References cumul, elapsed, end_t, getFractionalSeconds(), and start_t.

Referenced by main().

Here is the call graph for this function:

6.774.4 Member Data Documentation

6.774.4.1 cumul

```
double Timer::cumul  [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by CumulativeTime(), Reset(), and Stop().

6.774.4.2 elapsed

```
double Timer::elapsed  [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by ElapsedTime(), Reset(), and Stop().

6.774.4.3 end_t

```
double Timer::end_t  [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by Reset(), and Stop().

6.774.4.4 start_t

```
double Timer::start_t  [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 47 of file Timer.h.

Referenced by Reset(), Start(), and Stop().

6.774.4.5 sys_time

```
Function Timer::sys_time [private]
```

Examples

SugarPerformance/Timer.h.

Definition at line 46 of file Timer.h.

Referenced by getFractionalSeconds().

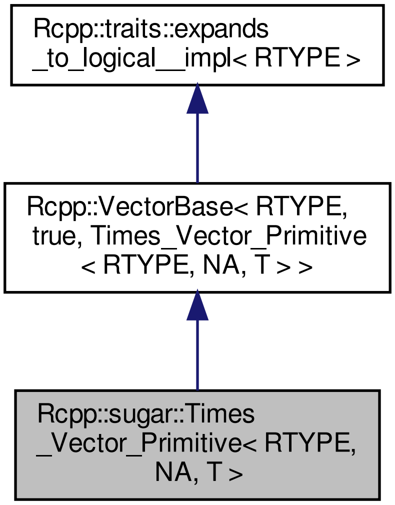
The documentation for this class was generated from the following file:

- inst/examples/SugarPerformance/Timer.h

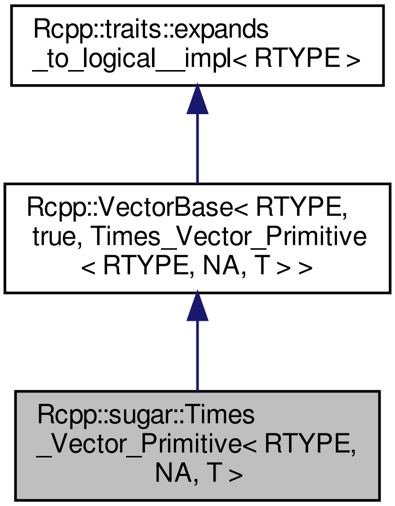
6.775 Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive < RTYPE, NA, T > >
Rcpp::sugar::Times _Vector_Primitive< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Times_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs
- bool rhs_na

6.775.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >

Definition at line 236 of file times.h.

6.775.2 Member Typedef Documentation

6.775.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE,
NA, T >::EXT
```

Definition at line 241 of file times.h.

6.775.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >↩
::STORAGE
```

Definition at line 239 of file times.h.

6.775.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 238 of file times.h.

6.775.3 Constructor & Destructor Documentation

6.775.3.1 Times_Vector_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::Times_Vector_Primitive (
             const VEC_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 243 of file times.h.

6.775.4 Member Function Documentation

6.775.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 247 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.775.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 253 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs.

6.775.5 Member Data Documentation

6.775.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs  [private]
```

Definition at line 256 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::size(), Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::size(), Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::size(), and Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::size().

6.775.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs  [private]
```

Definition at line 257 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::operator[ ]().

6.775.5.3 rhs_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs_na  [private]
```

Definition at line 258 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[ ]().

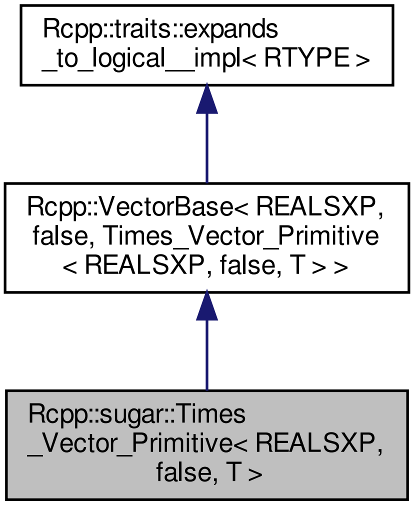
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

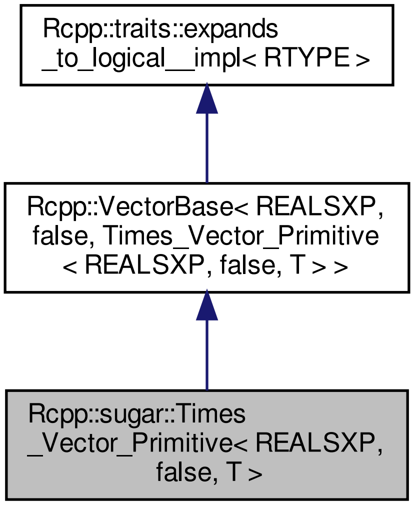
6.776 Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive < REALSXP, false, T > >
Rcpp::sugar::Times _Vector_Primitive< REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive < REALSXP, false, T > >
Rcpp::sugar::Times _Vector_Primitive< REALSXP, false, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, T >::type EXT

Public Member Functions

- Times_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.776.1 Detailed Description

template<typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >

Definition at line 309 of file times.h.

6.776.2 Member Typedef Documentation

6.776.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP, false, T>::type Rcpp::sugar::Times_Vector_Primitive<
REALSXP, false, T >::EXT
```

Definition at line 313 of file times.h.

6.776.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 312 of file times.h.

6.776.3 Constructor & Destructor Documentation

6.776.3.1 Times_Vector_Primitive()

```
template<typename T >
Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::Times_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 315 of file times.h.

6.776.4 Member Function Documentation

6.776.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 318 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs.

6.776.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 322 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs.

6.776.5 Member Data Documentation

6.776.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 325 of file times.h.

6.776.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 326 of file times.h.

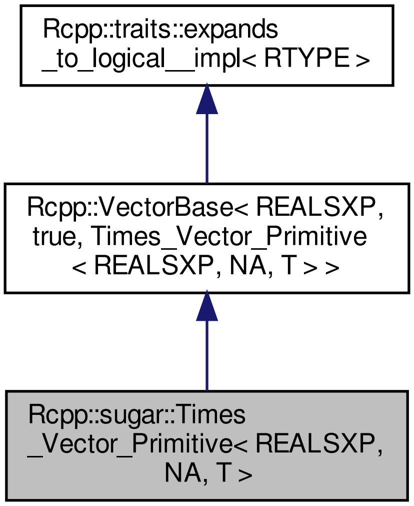
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

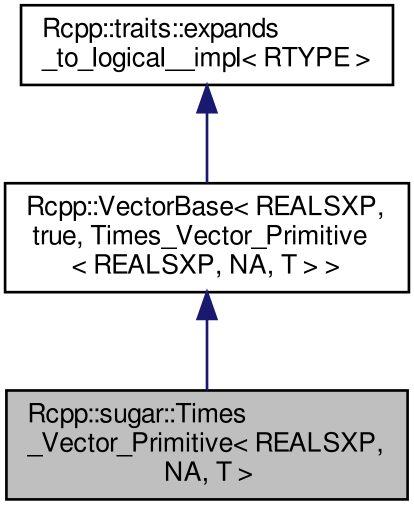
6.777 Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Times_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Times
_Vector_Primitive< REALSXP,
NA, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Times_Vector_Primitive
< REALSXP, NA, T > >
Rcpp::sugar::Times
_Vector_Primitive< REALSXP,
NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type EXT

Public Member Functions

- Times_Vector_Primitive (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.777.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >

Definition at line 262 of file times.h.

6.777.2 Member Typedef Documentation

6.777.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP, NA, T>::type Rcpp::sugar::Times_Vector_Primitive< REALSXP,
NA, T >::EXT
```

Definition at line 266 of file times.h.

6.777.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >↩
::VEC_TYPE
```

Definition at line 265 of file times.h.

6.777.3 Constructor & Destructor Documentation

6.777.3.1 Times_Vector_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::Times_Vector_Primitive (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 268 of file times.h.

6.777.4 Member Function Documentation

6.777.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 272 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs.

6.777.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 276 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs.

6.777.5 Member Data Documentation

6.777.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::lhs  [private]
```

Definition at line 279 of file times.h.

6.777.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::rhs  [private]
```

Definition at line 280 of file times.h.

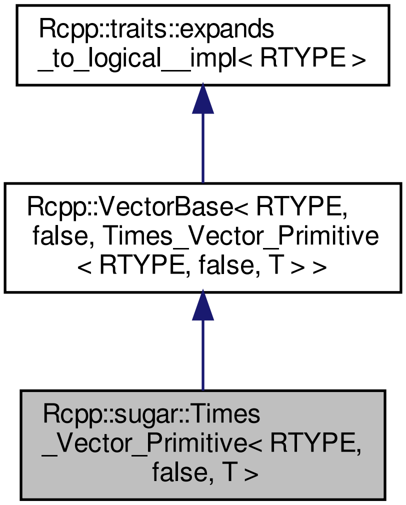
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

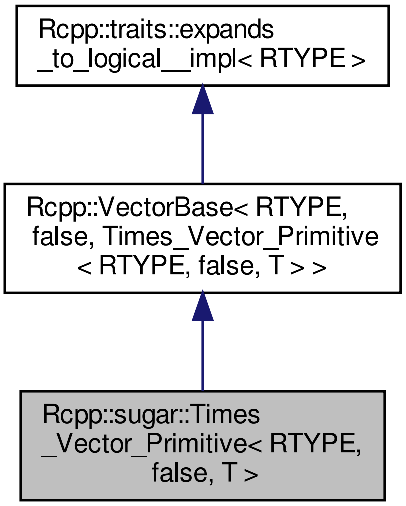
6.778 Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive < RTYPE, false, T > >
Rcpp::sugar::Times _Vector_Primitive< RTYPE, false, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type EXT

Public Member Functions

- Times_Vector_Primitive (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs
- bool rhs_na

6.778.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >

Definition at line 285 of file times.h.

6.778.2 Member Typedef Documentation

6.778.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Times_Vector_Primitive<
RTYPE, false, T >::EXT
```

Definition at line 290 of file times.h.

6.778.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 288 of file times.h.

6.778.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >↩
::VEC_TYPE
```

Definition at line 287 of file times.h.

6.778.3 Constructor & Destructor Documentation

6.778.3.1 Times_Vector_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::Times_Vector_Primitive (
            const VEC_TYPE & lhs_,
            STORAGE rhs_ )  [inline]
```

Definition at line 292 of file times.h.

6.778.4 Member Function Documentation

6.778.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 295 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs, Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs, and Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs_na.

6.778.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 299 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs.

6.778.5 Member Data Documentation

6.778.5.1 lhs

```
template<int RTYPE, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::lhs  [private]
```

Definition at line 302 of file times.h.

6.778.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::rhs  [private]
```

Definition at line 303 of file times.h.

6.778.5.3 rhs_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::rhs_na  [private]
```

Definition at line 304 of file times.h.

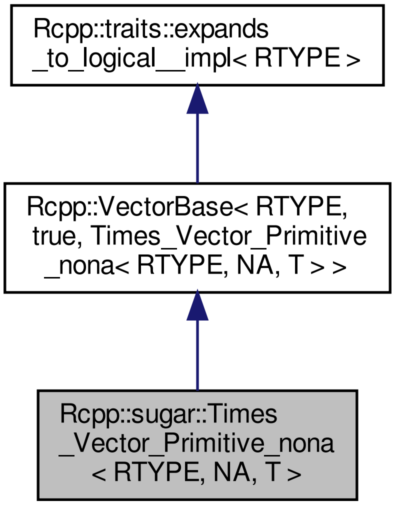
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

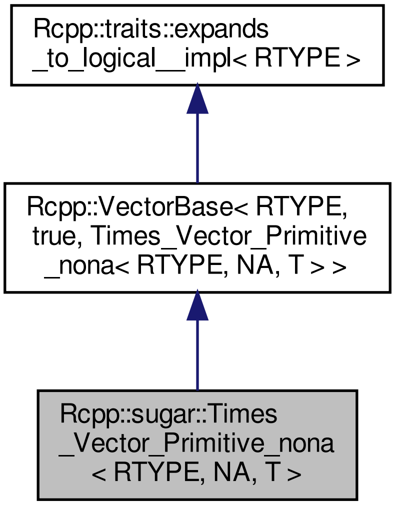
6.779 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive_nona< RTYPE, NA, T > >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, NA, T >::type EXT

Public Member Functions

- Times_Vector_Primitive_nona (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs

6.779.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >

Definition at line 336 of file times.h.

6.779.2 Member Typedef Documentation

6.779.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
RTYPE, NA, T >::EXT
```

Definition at line 340 of file times.h.

6.779.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T
>::STORAGE
```

Definition at line 339 of file times.h.

6.779.2.3 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >↩
::VEC_TYPE
```

Definition at line 338 of file times.h.

6.779.3 Constructor & Destructor Documentation

6.779.3.1 Times_Vector_Primitive_nona()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::Times_Vector_Primitive_nona (
                const VEC_TYPE & lhs_,
                STORAGE rhs_ )  [inline]
```

Definition at line 342 of file times.h.

6.779.4 Member Function Documentation

6.779.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 346 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs.

6.779.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 351 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.779.5 Member Data Documentation

6.779.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs  [private]
```

Definition at line 354 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::size(), Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::size(), Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::size(), and Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::size().

6.779.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs [private]
```

Definition at line 355 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::operator[ ](), Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::operator[ ](), and Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::operator[ ]().

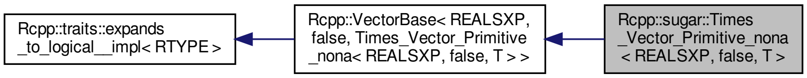
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

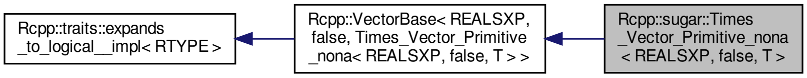
6.780 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive _nona< REALSXP, false, T > >
Rcpp::sugar::Times _Vector_Primitive_nona < REALSXP, false, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive _nona< REALSXP, false, T > >
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

6.780.1 Detailed Description

template<typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >

Definition at line 405 of file times.h.

6.780.2 Member Typedef Documentation

6.780.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP, false, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
REALSXP, false, T >::EXT
```

Definition at line 409 of file times.h.

6.780.2.2 VEC_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP,
false, T >::VEC_TYPE
```

Definition at line 408 of file times.h.

6.780.3 Constructor & Destructor Documentation

6.780.3.1 Times_Vector_Primitive_nona()

```
template<typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::Times_Vector_Primitive_nona (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 411 of file times.h.

6.780.4 Member Function Documentation

6.780.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 414 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_↩ Primitive_nona< RTYPE, NA, T >::rhs.

6.780.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::size ( ) const  [inline]
```

Definition at line 418 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.780.5 Member Data Documentation

6.780.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::lhs  [private]
```

Definition at line 421 of file times.h.

6.780.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::rhs  [private]
```

Definition at line 422 of file times.h.

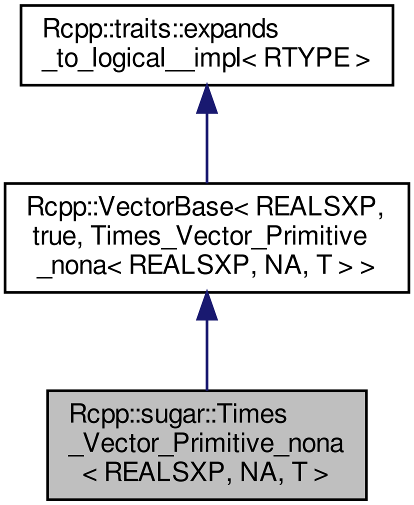
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

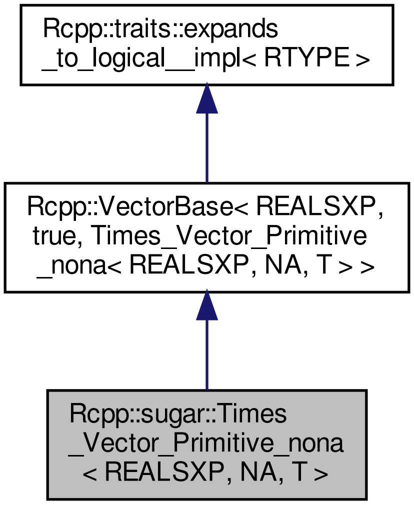
6.781 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >:


Rcpp::traits::expands
_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP,
true, Times_Vector_Primitive
_nona< REALSXP, NA, T > >
Rcpp::sugar::Times
_Vector_Primitive_nona
< REALSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive _nona< REALSXP, NA, T > >
Rcpp::sugar::Times _Vector_Primitive_nona < REALSXP, NA, T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, T > VEC_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, NA, T >::type EXT

Public Member Functions

- Times_Vector_Primitive_nona (const VEC_TYPE &lhs_, double rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- double rhs

6.781.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >

Definition at line 360 of file times.h.

6.781.2 Member Typedef Documentation

6.781.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
REALSXP, NA, T >::EXT
```

Definition at line 364 of file times.h.

6.781.2.2 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T
>::VEC_TYPE
```

Definition at line 363 of file times.h.

6.781.3 Constructor & Destructor Documentation

6.781.3.1 Times_Vector_Primitive_nona()

```
template<bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::Times_Vector_Primitive_nona (
             const VEC_TYPE & lhs_,
             double rhs_ )  [inline]
```

Definition at line 366 of file times.h.

6.781.4 Member Function Documentation

6.781.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 370 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_↩ Primitive_nona< RTYPE, NA, T >::rhs.

6.781.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::size ( ) const  [inline]
```

Definition at line 374 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.781.5 Member Data Documentation

6.781.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::lhs  [private]
```

Definition at line 377 of file times.h.

6.781.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::rhs  [private]
```

Definition at line 378 of file times.h.

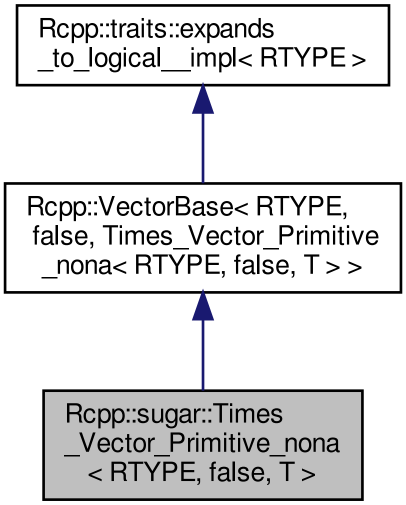
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

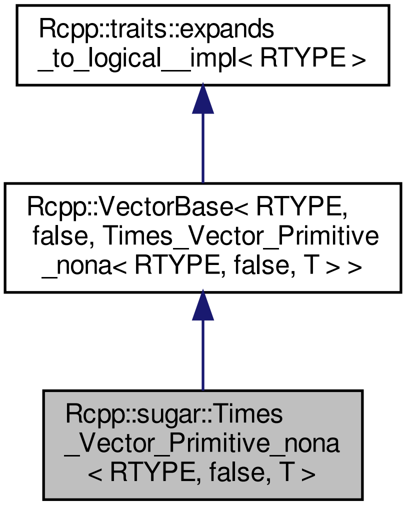
6.782 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive_nona< RTYPE, false, T > >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >

Collaboration diagram for Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive _nona< RTYPE, false, T > >
Rcpp::sugar::Times _Vector_Primitive_nona < RTYPE, false, T >

Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, T >::type EXT

Public Member Functions

- Times_Vector_Primitive_nona (const VEC_TYPE &lhs_, STORAGE rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const EXT & lhs
- STORAGE rhs

6.782.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >

Definition at line 383 of file times.h.

6.782.2 Member Typedef Documentation

6.782.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
RTYPE, false, T >::EXT
```

Definition at line 388 of file times.h.

6.782.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false,
T >::STORAGE
```

Definition at line 386 of file times.h.

6.782.2.3 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 385 of file times.h.

6.782.3 Constructor & Destructor Documentation

6.782.3.1 Times_Vector_Primitive_nona()

```
template<int RTYPE, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::Times_Vector_Primitive_nona (
             const VEC_TYPE & lhs_,
             STORAGE rhs_ )  [inline]
```

Definition at line 390 of file times.h.

6.782.4 Member Function Documentation

6.782.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 393 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times_Vector_↩ Primitive_nona< RTYPE, NA, T >::rhs.

6.782.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::size ( ) const  [inline]
```

Definition at line 397 of file times.h.

References Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs.

6.782.5 Member Data Documentation

6.782.5.1 lhs

```
template<int RTYPE, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::lhs  [private]
```

Definition at line 400 of file times.h.

6.782.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::rhs  [private]
```

Definition at line 401 of file times.h.

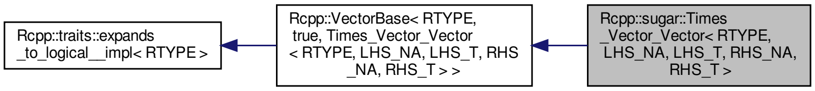
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

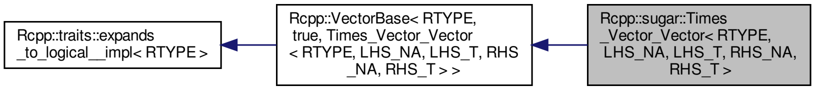
6.783 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.783.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 29 of file times.h.

6.783.2 Member Typedef Documentation

6.783.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file times.h.

6.783.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file times.h.

6.783.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file times.h.

6.783.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file times.h.

6.783.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 31 of file times.h.

6.783.3 Constructor & Destructor Documentation

6.783.3.1 Times_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 38 of file times.h.

6.783.4 Member Function Documentation

6.783.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 42 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.783.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 49 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.783.5 Member Data Documentation

6.783.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 52 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size(), and Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size().

6.783.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 53 of file times.h.

Referenced by Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[ ](), Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[ ](), and Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[ ]().

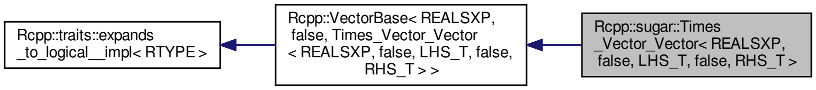
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

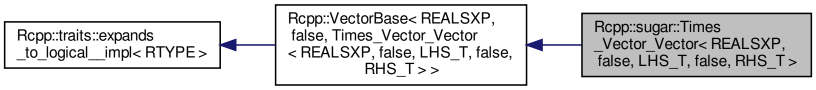
6.784 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.784.1 Detailed Description

template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >

Definition at line 211 of file times.h.

6.784.2 Member Typedef Documentation

6.784.2.1 LHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 217 of file times.h.

6.784.2.2 LHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 214 of file times.h.

6.784.2.3 RHS_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 218 of file times.h.

6.784.2.4 RHS_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 215 of file times.h.

6.784.3 Constructor & Destructor Documentation

6.784.3.1 Times_Vector_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 220 of file times.h.

6.784.4 Member Function Documentation

6.784.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 223 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.784.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 227 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.784.5 Member Data Documentation

6.784.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 230 of file times.h.

6.784.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 231 of file times.h.

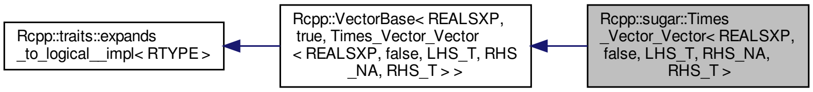
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

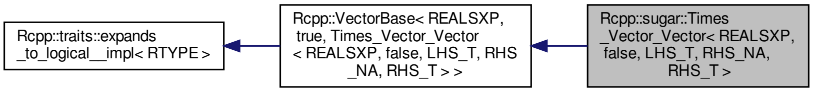
6.785 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.785.1 Detailed Description

template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >

Definition at line 110 of file times.h.

6.785.2 Member Typedef Documentation

6.785.2.1 LHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 116 of file times.h.

6.785.2.2 LHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 113 of file times.h.

6.785.2.3 RHS_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 117 of file times.h.

6.785.2.4 RHS_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 114 of file times.h.

6.785.3 Constructor & Destructor Documentation

6.785.3.1 Times_Vector_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 119 of file times.h.

6.785.4 Member Function Documentation

6.785.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 122 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.785.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 126 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.785.5 Member Data Documentation

6.785.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 129 of file times.h.

6.785.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 130 of file times.h.

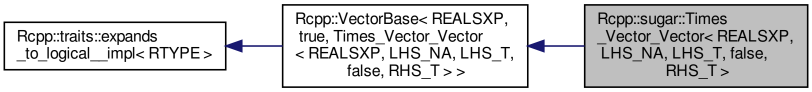
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

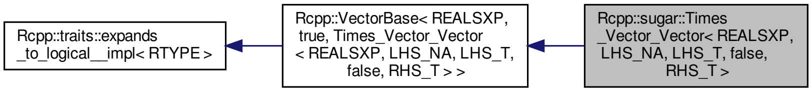
6.786 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector < REALSXP, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector < REALSXP, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Times _Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, false, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.786.1 Detailed Description

template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >

Definition at line 162 of file times.h.

6.786.2 Member Typedef Documentation

6.786.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 168 of file times.h.

6.786.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 165 of file times.h.

6.786.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 169 of file times.h.

6.786.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 166 of file times.h.

6.786.3 Constructor & Destructor Documentation

6.786.3.1 Times_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 171 of file times.h.

6.786.4 Member Function Documentation

6.786.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 174 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.786.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 178 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.786.5 Member Data Documentation

6.786.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 181 of file times.h.

6.786.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 182 of file times.h.

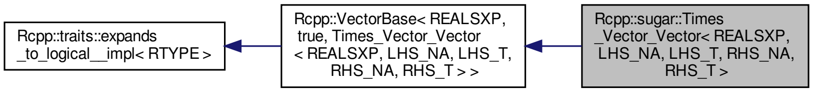
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

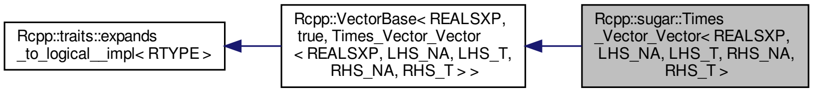
6.787 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector < REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Times _Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< REALSXP, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< REALSXP, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< REALSXP, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.787.1 Detailed Description

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 57 of file times.h.

6.787.2 Member Typedef Documentation

6.787.2.1 LHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 63 of file times.h.

6.787.2.2 LHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 60 of file times.h.

6.787.2.3 RHS_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 64 of file times.h.

6.787.2.4 RHS_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 61 of file times.h.

6.787.3 Constructor & Destructor Documentation

6.787.3.1 Times_Vector_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 66 of file times.h.

6.787.4 Member Function Documentation

6.787.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 70 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp↩
::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.787.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const  [inline]
```

Definition at line 74 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.787.5 Member Data Documentation

6.787.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 77 of file times.h.

6.787.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 78 of file times.h.

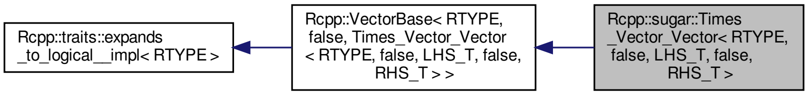
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

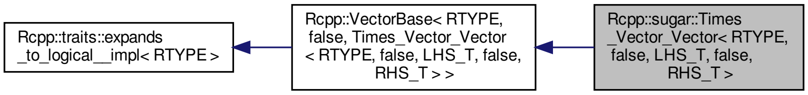
6.788 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector < RTYPE, false, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector < RTYPE, false, LHS_T, false, RHS_T > >
Rcpp::sugar::Times _Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.788.1 Detailed Description

template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >

Definition at line 187 of file times.h.

6.788.2 Member Typedef Documentation

6.788.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 193 of file times.h.

6.788.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS↩
_T, false, RHS_T >::LHS_TYPE
```

Definition at line 190 of file times.h.

6.788.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 194 of file times.h.

6.788.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 191 of file times.h.

6.788.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 189 of file times.h.

6.788.3 Constructor & Destructor Documentation

6.788.3.1 Times_Vector_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Times_Vector_Vector (
            const LHS_TYPE & lhs_,
            const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 196 of file times.h.

6.788.4 Member Function Documentation

6.788.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
            R_xlen_t i ) const  [inline]
```

Definition at line 199 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.788.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 203 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.788.5 Member Data Documentation

6.788.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 206 of file times.h.

6.788.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 207 of file times.h.

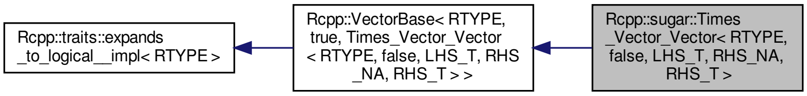
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

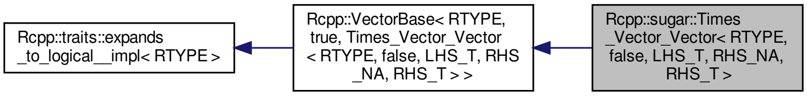
6.789 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >
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 >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > >
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, false, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.789.1 Detailed Description

template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >

Definition at line 84 of file times.h.

6.789.2 Member Typedef Documentation

6.789.2.1 LHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 90 of file times.h.

6.789.2.2 LHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS↩
_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 87 of file times.h.

6.789.2.3 RHS_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 91 of file times.h.

6.789.2.4 RHS_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 88 of file times.h.

6.789.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 86 of file times.h.

6.789.3 Constructor & Destructor Documentation

6.789.3.1 Times_Vector_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
                const LHS_TYPE & lhs_,
                const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 93 of file times.h.

6.789.4 Member Function Documentation

6.789.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
                R_xlen_t i ) const  [inline]
```

Definition at line 96 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.789.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 102 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.789.5 Member Data Documentation

6.789.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs  [private]
```

Definition at line 105 of file times.h.

6.789.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs  [private]
```

Definition at line 106 of file times.h.

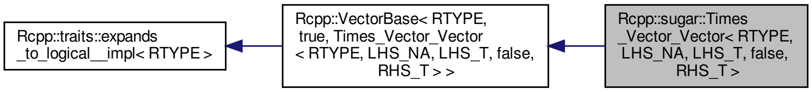
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

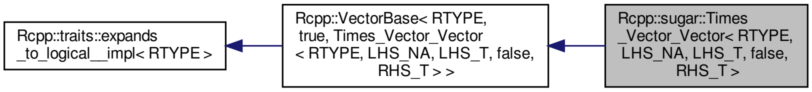
6.790 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
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 >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector < RTYPE, LHS_NA, LHS_T, false, RHS_T > >
Rcpp::sugar::Times _Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > LHS_TYPE
- typedef Rcpp::VectorBase< RTYPE, false, RHS_T > RHS_TYPE
- typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T >::type LHS_EXT
- typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T >::type RHS_EXT

Public Member Functions

- Times_Vector_Vector (const LHS_TYPE &lhs_, const RHS_TYPE &rhs_)
- STORAGE operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const LHS_EXT & lhs
- const RHS_EXT & rhs

6.790.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >

Definition at line 136 of file times.h.

6.790.2 Member Typedef Documentation

6.790.2.1 LHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 142 of file times.h.

6.790.2.2 LHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 139 of file times.h.

6.790.2.3 RHS_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 143 of file times.h.

6.790.2.4 RHS_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 140 of file times.h.

6.790.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 138 of file times.h.

6.790.3 Constructor & Destructor Documentation

6.790.3.1 Times_Vector_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Times_Vector_Vector (
             const LHS_TYPE & lhs_,
             const RHS_TYPE & rhs_ )  [inline]
```

Definition at line 145 of file times.h.

6.790.4 Member Function Documentation

6.790.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 148 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs, and Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs.

6.790.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 154 of file times.h.

References Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs.

6.790.5 Member Data Documentation

6.790.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs  [private]
```

Definition at line 157 of file times.h.

6.790.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs  [private]
```

Definition at line 158 of file times.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

6.791 Rcpp::ttinfo Struct Reference

Public Attributes

- int_fast32_t tt_gmtoff
- int tt_isdst
- int tt_abbrind
- int tt_ttisstd
- int tt_ttisgmt

6.791.1 Detailed Description

Definition at line 380 of file date.cpp.

6.791.2 Member Data Documentation

6.791.2.1 tt_abbrind

```
int Rcpp::ttinfo::tt_abbrind
```

Definition at line 383 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.791.2.2 tt_gmtoff

```
int_fast32_t Rcpp::ttinfo::tt_gmtoff
```

Definition at line 381 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.791.2.3 tt_isdst

```
int Rcpp::ttinfo::tt_isdst
```

Definition at line 382 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.791.2.4 tt_ttisgmt

```
int Rcpp::ttinfo::tt_ttisgmt
```

Definition at line 385 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

6.791.2.5 tt_ttisstd

```
int Rcpp::ttinfo::tt_ttisstd
```

Definition at line 384 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

The documentation for this struct was generated from the following file:

- src/date.cpp

6.792 Rcpp::attributes::Type Class Reference

Public Member Functions

- Type ()
- Type (const std::string &name, bool isConst, bool isReference)
- bool empty () const
- bool operator== (const Type &other) const
- bool operator!= (const Type &other) const
- const std::string & name () const
- std::string full_name () const
- bool isVoid () const
- bool isConst () const
- bool isReference () const

Private Attributes

- std::string name_
- bool isConst_
- bool isReference_

6.792.1 Detailed Description

Definition at line 172 of file attributes.cpp.

6.792.2 Constructor & Destructor Documentation

6.792.2.1 Type() [1/2]

```
Rcpp::attributes::Type::Type ( )  [inline]
```

Definition at line 174 of file attributes.cpp.

6.792.2.2 Type() [2/2]

```
Rcpp::attributes::Type::Type (
             const std::string & name,
             bool isConst,
             bool isReference )  [inline]
```

Definition at line 175 of file attributes.cpp.

6.792.3 Member Function Documentation

6.792.3.1 empty()

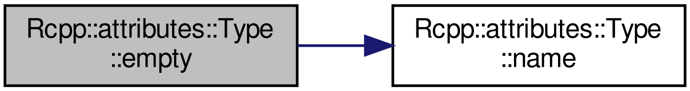
```
bool Rcpp::attributes::Type::empty ( ) const  [inline]
```

Definition at line 179 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::Argument::empty(), and Rcpp::attributes::operator<<().

Here is the call graph for this function:



6.792.3.2 full_name()

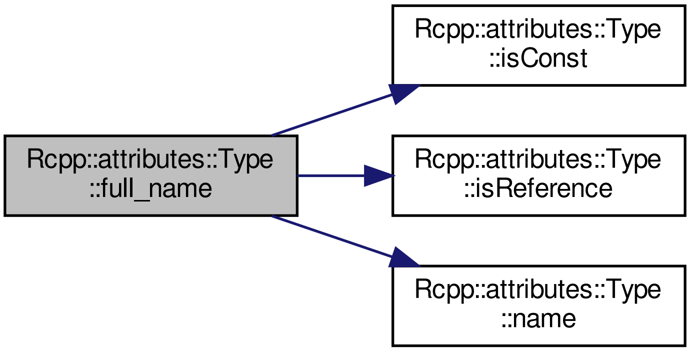
```
std::string Rcpp::attributes::Type::full_name ( ) const  [inline]
```

Definition at line 192 of file attributes.cpp.

References isConst(), isReference(), and name().

Referenced by Rcpp::attributes::generateCpp().

Here is the call graph for this function:



6.792.3.3 isConst()

```
bool Rcpp::attributes::Type::isConst ( ) const  [inline]
```

Definition at line 201 of file attributes.cpp.

References isConst_.

Referenced by full_name(), and Rcpp::attributes::operator<<().

6.792.3.4 isReference()

```
bool Rcpp::attributes::Type::isReference ( ) const  [inline]
```

Definition at line 202 of file attributes.cpp.

References isReference_.

Referenced by full_name(), and Rcpp::attributes::operator<<().

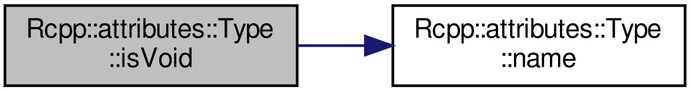
6.792.3.5 isVoid()

```
bool Rcpp::attributes::Type::isVoid ( ) const  [inline]
```

Definition at line 200 of file attributes.cpp.

References name().

Here is the call graph for this function:


Rcpp::attributes::Type ::isVoid
Rcpp::attributes::Type ::name

6.792.3.6 name()

```
const std::string& Rcpp::attributes::Type::name ( ) const  [inline]
```

Definition at line 191 of file attributes.cpp.

References name_.

Referenced by empty(), full_name(), Rcpp::attributes::generateRArgList(), isVoid(), and Rcpp::attributes::operator<<().

6.792.3.7 operator"!=()

```
bool Rcpp::attributes::Type::operator!= (
             const Type & other ) const  [inline]
```

Definition at line 187 of file attributes.cpp.

6.792.3.8 operator==()

```
bool Rcpp::attributes::Type::operator== (
             const Type & other ) const  [inline]
```

Definition at line 181 of file attributes.cpp.

References isConst_, isReference_, and name_.

6.792.4 Member Data Documentation

6.792.4.1 isConst_

```
bool Rcpp::attributes::Type::isConst_  [private]
```

Definition at line 206 of file attributes.cpp.

Referenced by isConst(), and operator==().

6.792.4.2 isReference_

```
bool Rcpp::attributes::Type::isReference_  [private]
```

Definition at line 207 of file attributes.cpp.

Referenced by isReference(), and operator==().

6.792.4.3 name_

```
std::string Rcpp::attributes::Type::name_  [private]
```

Definition at line 205 of file attributes.cpp.

Referenced by name(), and operator==().

The documentation for this class was generated from the following file:

- src/attributes.cpp

6.793 Rcpp::tzhead Struct Reference

Public Attributes

- char tzh_magic [4]
- char tzh_version [1]
- char tzh_reserved [15]
- char tzh_ttisgmtcnt [4]
- char tzh_ttisstdcnt [4]
- char tzh_leapcnt [4]
- char tzh_timecnt [4]
- char tzh_typecnt [4]
- char tzh_charcnt [4]

6.793.1 Detailed Description

Definition at line 205 of file date.cpp.

6.793.2 Member Data Documentation

6.793.2.1 tzh_charcnt

```
char Rcpp::tzhead::tzh_charcnt[4]
```

Definition at line 214 of file date.cpp.

6.793.2.2 tzh_leapcnt

```
char Rcpp::tzhead::tzh_leapcnt[4]
```

Definition at line 211 of file date.cpp.

6.793.2.3 tzh_magic

```
char Rcpp::tzhead::tzh_magic[4]
```

Definition at line 206 of file date.cpp.

6.793.2.4 tzh_reserved

```
char Rcpp::tzhead::tzh_reserved[15]
```

Definition at line 208 of file date.cpp.

6.793.2.5 tzh_timecnt

```
char Rcpp::tzhead::tzh_timecnt[4]
```

Definition at line 212 of file date.cpp.

6.793.2.6 tzh_ttisgmtcnt

```
char Rcpp::tzhead::tzh_ttisgmtcnt[4]
```

Definition at line 209 of file date.cpp.

6.793.2.7 tzh_ttisstdcnt

```
char Rcpp::tzhead::tzh_ttisstdcnt[4]
```

Definition at line 210 of file date.cpp.

6.793.2.8 tzh_typecnt

```
char Rcpp::tzhead::tzh_typecnt[4]
```

Definition at line 213 of file date.cpp.

6.793.2.9 tzh_version

```
char Rcpp::tzhead::tzh_version[1]
```

Definition at line 207 of file date.cpp.

The documentation for this struct was generated from the following file:

- src/date.cpp

6.794 Rcpp::traits::un_pointer< T > Struct Template Reference

```
#include <un_pointer.h>
```

Public Types

- typedef T type

6.794.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< T >

Definition at line 29 of file un_pointer.h.

6.794.2 Member Typedef Documentation

6.794.2.1 type

```
template<typename T >
typedef T Rcpp::traits::un_pointer< T >::type
```

Definition at line 29 of file un_pointer.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un_pointer.h

6.795 Rcpp::traits::un_pointer< object< T > > Struct Template Reference

```
#include <un_pointer.h>
```

Public Types

- typedef T type

6.795.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< object< T > >

Definition at line 31 of file un_pointer.h.

6.795.2 Member Typedef Documentation

6.795.2.1 type

```
template<typename T >
typedef T Rcpp::traits::un_pointer< object< T > >::type
```

Definition at line 31 of file un_pointer.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un_pointer.h

6.796 Rcpp::traits::un_pointer< T ∗ > Struct Template Reference

```
#include <un_pointer.h>
```

Public Types

- typedef T type

6.796.1 Detailed Description

template<typename T>
struct Rcpp::traits::un_pointer< T ∗ >

Definition at line 30 of file un_pointer.h.

6.796.2 Member Typedef Documentation

6.796.2.1 type

```
template<typename T >
typedef T Rcpp::traits::un_pointer< T * >::type
```

Definition at line 30 of file un_pointer.h.

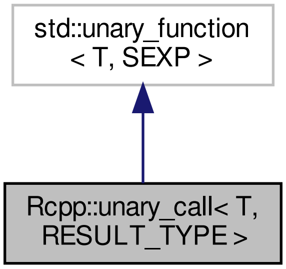
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un_pointer.h

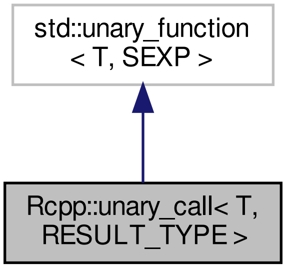
6.797 Rcpp::unary_call< T, RESULT_TYPE > Class Template Reference

```
#include <Language.h>
```

Inheritance diagram for Rcpp::unary_call< T, RESULT_TYPE >:



Collaboration diagram for Rcpp::unary_call< T, RESULT_TYPE >:



Public Member Functions

- unary_call (Language call_)
- unary_call (Language call_, R_xlen_t index)
- unary_call (Function fun)
- RESULT_TYPE operator() (const T &object)

Private Attributes

- Language call
- Language::Proxy proxy

6.797.1 Detailed Description

template<typename T, typename RESULT_TYPE = SEXP>
class Rcpp::unary_call< T, RESULT_TYPE >

Definition at line 180 of file Language.h.

6.797.2 Constructor & Destructor Documentation

6.797.2.1 unary_call() [1/3]

```
template<typename T , typename RESULT_TYPE = SEXP>
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (
            Language call_ )  [inline]
```

Definition at line 185 of file Language.h.

6.797.2.2 unary_call() [2/3]

```
template<typename T , typename RESULT_TYPE = SEXP>
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (
            Language call_,
            R_xlen_t index )  [inline]
```

Definition at line 186 of file Language.h.

6.797.2.3 unary_call() [3/3]

```
template<typename T , typename RESULT_TYPE = SEXP>
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (
            Function fun )  [inline]
```

Definition at line 187 of file Language.h.

6.797.3 Member Function Documentation

6.797.3.1 operator()()

```
template<typename T , typename RESULT_TYPE = SEXP>
RESULT_TYPE Rcpp::unary_call< T, RESULT_TYPE >::operator() (
             const T & object )  [inline]
```

Definition at line 189 of file Language.h.

References Rcpp::unary_call< T, RESULT_TYPE >::call, and Rcpp::unary_call< T, RESULT_TYPE >::proxy.

6.797.4 Member Data Documentation

6.797.4.1 call

```
template<typename T , typename RESULT_TYPE = SEXP>
Language Rcpp::unary_call< T, RESULT_TYPE >::call  [private]
```

Definition at line 195 of file Language.h.

Referenced by Rcpp::unary_call< T, RESULT_TYPE >::operator()(), and Rcpp::binary_call< T1, T2, RESULT_TYPE >::operator()().

6.797.4.2 proxy

```
template<typename T , typename RESULT_TYPE = SEXP>
Language::Proxy Rcpp::unary_call< T, RESULT_TYPE >::proxy  [private]
```

Definition at line 196 of file Language.h.

Referenced by Rcpp::unary_call< T, RESULT_TYPE >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Language.h

6.798 Rcpp::sugar::unary_minus< RTYPE, NA > Class Template Reference

```
#include <unary_minus.h>
```

Public Types

- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef unary_minus_result_type< RTYPE >::type RESULT

Public Member Functions

- RESULT apply (STORAGE x) const

6.798.1 Detailed Description

template<int RTYPE, bool NA>
class Rcpp::sugar::unary_minus< RTYPE, NA >

Definition at line 41 of file unary_minus.h.

6.798.2 Member Typedef Documentation

6.798.2.1 RESULT

```
template<int RTYPE, bool NA>
typedef unary_minus_result_type<RTYPE>::type Rcpp::sugar::unary_minus< RTYPE, NA >::RESULT
```

Definition at line 44 of file unary_minus.h.

6.798.2.2 STORAGE

```
template<int RTYPE, bool NA>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::unary_minus< RTYPE, NA >::STORAGE
```

Definition at line 43 of file unary_minus.h.

6.798.3 Member Function Documentation

6.798.3.1 apply()

```
template<int RTYPE, bool NA>
RESULT Rcpp::sugar::unary_minus< RTYPE, NA >::apply (
             STORAGE x ) const [inline]
```

Definition at line 45 of file unary_minus.h.

Referenced by Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::operator[ ]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

6.799 Rcpp::sugar::unary_minus< CPLXSXP, false > Class Reference

```
#include <unary_minus.h>
```

Public Member Functions

- Rcomplex apply (Rcomplex x) const

6.799.1 Detailed Description

Definition at line 71 of file unary_minus.h.

6.799.2 Member Function Documentation

6.799.2.1 apply()

```
Rcomplex Rcpp::sugar::unary_minus< 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

6.800 Rcpp::sugar::unary_minus< CPLXSXP, NA > Class Template Reference

```
#include <unary_minus.h>
```

Public Member Functions

- Rcomplex apply (Rcomplex x) const

6.800.1 Detailed Description

template<bool NA>
class Rcpp::sugar::unary_minus< CPLXSXP, NA >

Definition at line 59 of file unary_minus.h.

6.800.2 Member Function Documentation

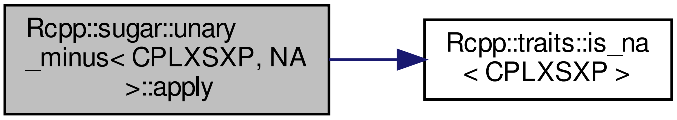
6.800.2.1 apply()

```
template<bool NA>
Rcomplex Rcpp::sugar::unary_minus< 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

6.801 Rcpp::sugar::unary_minus< RTYPE, false > Class Template Reference

```
#include <unary_minus.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE
- typedef unary_minus_result_type< RTYPE >::type RESULT

Public Member Functions

- RESULT apply (STORAGE x) const

6.801.1 Detailed Description

template<int RTYPE>
class Rcpp::sugar::unary_minus< RTYPE, false >

Definition at line 50 of file unary_minus.h.

6.801.2 Member Typedef Documentation

6.801.2.1 RESULT

```
template<int RTYPE>
typedef unary_minus_result_type<RTYPE>::type Rcpp::sugar::unary_minus< RTYPE, false >::RESULT
```

Definition at line 53 of file unary_minus.h.

6.801.2.2 STORAGE

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::unary_minus< RTYPE, false >::STORAGE
```

Definition at line 52 of file unary_minus.h.

6.801.3 Member Function Documentation

6.801.3.1 apply()

```
template<int RTYPE>
RESULT Rcpp::sugar::unary_minus< RTYPE, false >::apply (
             STORAGE x ) const  [inline]
```

Definition at line 54 of file unary_minus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

6.802 Rcpp::sugar::unary_minus_result_type< RTYPE > Struct Template Reference

```
#include <unary_minus.h>
```

Public Types

- enum { value = RTYPE }
- typedef traits::storage_type< RTYPE >::type type

6.802.1 Detailed Description

template<int RTYPE>
struct Rcpp::sugar::unary_minus_result_type< RTYPE >

Definition at line 29 of file unary_minus.h.

6.802.2 Member Typedef Documentation

6.802.2.1 type

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::unary_minus_result_type< RTYPE >::type
```

Definition at line 30 of file unary_minus.h.

6.802.3 Member Enumeration Documentation

6.802.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

Enumerator

value	

Definition at line 31 of file unary_minus.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

6.803 Rcpp::sugar::unary_minus_result_type< LGLSXP > Struct Reference

```
#include <unary_minus.h>
```

Public Types

- enum { value = INTSXP }
- typedef traits::storage_type< INTSXP >::type type

6.803.1 Detailed Description

Definition at line 34 of file unary_minus.h.

6.803.2 Member Typedef Documentation

6.803.2.1 type

```
typedef traits::storage_type<INTSXP>::type Rcpp::sugar::unary_minus_result_type< LGLSXP >::type
```

Definition at line 35 of file unary_minus.h.

6.803.3 Member Enumeration Documentation

6.803.3.1 anonymous enum

```
anonymous enum
```

Enumerator

value	

Definition at line 36 of file unary_minus.h.

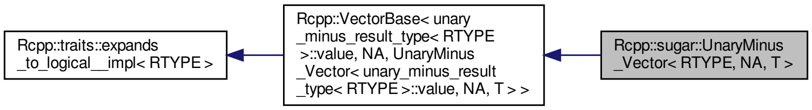
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

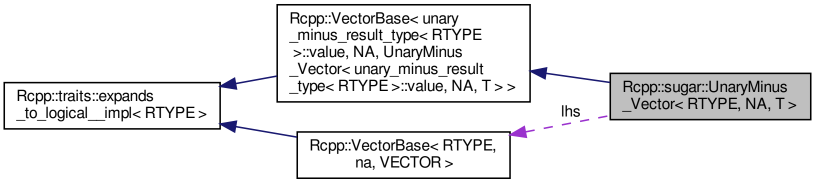
6.804 Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > Class Template Reference

```
#include <unary_minus.h>
```

Inheritance diagram for Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >:


Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< unary _minus_result_type< RTYPE >::value, NA, UnaryMinus _Vector< unary_minus_result _type< RTYPE >::value, NA, T > >
Rcpp::sugar::UnaryMinus _Vector< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >:


Rcpp::VectorBase< unary _minus_result_type< RTYPE >::value, NA, UnaryMinus _Vector< unary_minus_result _type< RTYPE >::value, NA, T > >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::sugar::UnaryMinus _Vector< RTYPE, NA, T >
lhs
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef traits::storage_type< RTYPE >::type STORAGE
- typedef unary_minus_result_type< RTYPE >::type RESULT
- typedef unary_minus< RTYPE, NA > OPERATOR

Public Member Functions

- UnaryMinus_Vector (const VEC_TYPE &lhs_)
- RESULT operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_TYPE & lhs
- OPERATOR op

6.804.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >

Definition at line 83 of file unary_minus.h.

6.804.2 Member Typedef Documentation

6.804.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >
typedef unary_minus<RTYPE,NA> Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::OPERATOR
```

Definition at line 92 of file unary_minus.h.

6.804.2.2 RESULT

```
template<int RTYPE, bool NA, typename T >
typedef unary_minus_result_type<RTYPE>::type Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >↩
::RESULT
```

Definition at line 91 of file unary_minus.h.

6.804.2.3 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::STORAGE
```

Definition at line 90 of file unary_minus.h.

6.804.2.4 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 89 of file unary_minus.h.

6.804.3 Constructor & Destructor Documentation

6.804.3.1 UnaryMinus_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::UnaryMinus_Vector (
             const VEC_TYPE & lhs_ )  [inline]
```

Definition at line 94 of file unary_minus.h.

6.804.4 Member Function Documentation

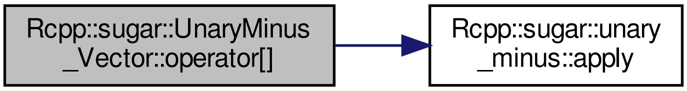
6.804.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
RESULT Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 97 of file unary_minus.h.

References Rcpp::sugar::unary_minus< RTYPE, NA >::apply(), Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::op.

Here is the call graph for this function:



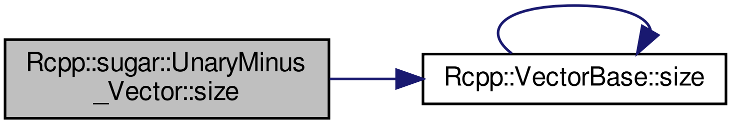
6.804.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 101 of file unary_minus.h.

References Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.804.5 Member Data Documentation

6.804.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::lhs  [private]
```

Definition at line 104 of file unary_minus.h.

Referenced by Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::operator[ ](), and Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::size().

6.804.5.2 op

```
template<int RTYPE, bool NA, typename T >
OPERATOR Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::op  [private]
```

Definition at line 105 of file unary_minus.h.

Referenced by Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::operator[ ]().

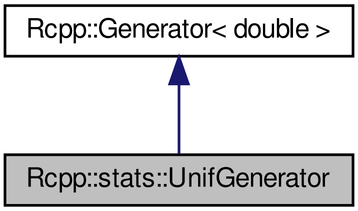
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary_minus.h

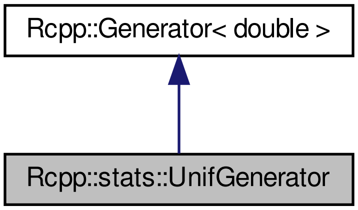
6.805 Rcpp::stats::UnifGenerator Class Reference

```
#include <runif.h>
```

Inheritance diagram for Rcpp::stats::UnifGenerator:



Collaboration diagram for Rcpp::stats::UnifGenerator:



Public Member Functions

- UnifGenerator (double min_=0.0, double max_=1.0)
- double operator() () const

Private Attributes

- double min
- double diff

Additional Inherited Members

6.805.1 Detailed Description

Definition at line 28 of file runif.h.

6.805.2 Constructor & Destructor Documentation

6.805.2.1 UnifGenerator()

```
Rcpp::stats::UnifGenerator::UnifGenerator (
             double min_ = 0.0,
             double max_ = 1.0 )  [inline]
```

Definition at line 31 of file runif.h.

6.805.3 Member Function Documentation

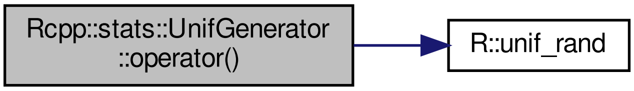
6.805.3.1 operator()()

```
double Rcpp::stats::UnifGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file runif.h.

References diff, min, and R::unif_rand().

Here is the call graph for this function:



6.805.4 Member Data Documentation

6.805.4.1 diff

```
double Rcpp::stats::UnifGenerator::diff  [private]
```

Definition at line 42 of file runif.h.

Referenced by operator()().

6.805.4.2 min

```
double Rcpp::stats::UnifGenerator::min  [private]
```

Definition at line 41 of file runif.h.

Referenced by operator()().

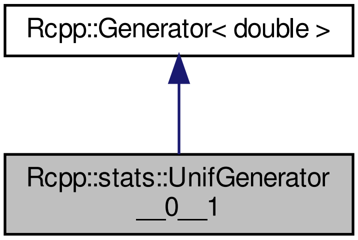
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/runif.h

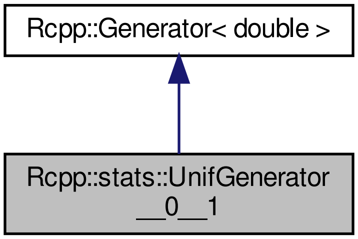
6.806 Rcpp::stats::UnifGenerator__0__1 Class Reference

```
#include <runif.h>
```

Inheritance diagram for Rcpp::stats::UnifGenerator__0__1:



Collaboration diagram for Rcpp::stats::UnifGenerator__0__1:



Public Member Functions

- UnifGenerator__0__1 ()
- double operator() () const

Additional Inherited Members

6.806.1 Detailed Description

Definition at line 45 of file runif.h.

6.806.2 Constructor & Destructor Documentation

6.806.2.1 UnifGenerator__0__1()

```
Rcpp::stats::UnifGenerator__0__1::UnifGenerator__0__1 ( )  [inline]
```

Definition at line 48 of file runif.h.

6.806.3 Member Function Documentation

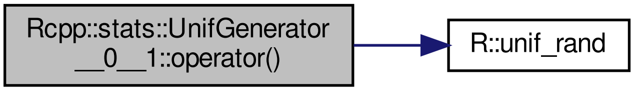
6.806.3.1 operator()()

```
double Rcpp::stats::UnifGenerator__0__1::operator() ( ) const  [inline]
```

Definition at line 50 of file runif.h.

References R::unif_rand().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/runif.h

6.807 Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference

```
#include <setdiff.h>
```

Public Types

- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- Union (const LHS_T &lhs, const RHS_T &rhs)
- Vector< RTYPE > get () const

Private Types

- typedef RCPP_UNORDERED_SET< STORAGE > SET
- typedef SET::const_iterator ITERATOR

Private Attributes

- SET result

6.807.1 Detailed Description

template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Definition at line 133 of file setdiff.h.

6.807.2 Member Typedef Documentation

6.807.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 152 of file setdiff.h.

6.807.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >↩
::SET  [private]
```

Definition at line 151 of file setdiff.h.

6.807.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T >::STORAGE
```

Definition at line 135 of file setdiff.h.

6.807.3 Constructor & Destructor Documentation

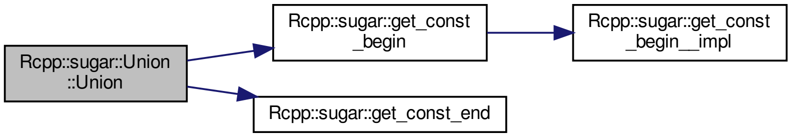
6.807.3.1 Union()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Union (
             const LHS_T & lhs,
             const RHS_T & rhs )  [inline]
```

Definition at line 137 of file setdiff.h.

References Rcpp::sugar::get_const_begin(), and Rcpp::sugar::get_const_end().

Here is the call graph for this function:



6.807.4 Member Function Documentation

6.807.4.1 get()

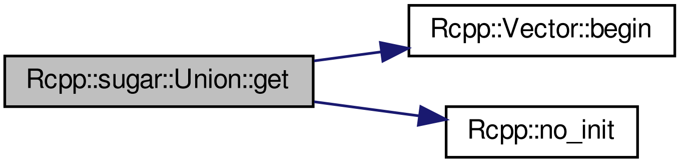
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const  [inline]
```

Definition at line 143 of file setdiff.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::no_init().

Referenced by Rcpp::union_().

Here is the call graph for this function:



6.807.5 Member Data Documentation

6.807.5.1 result

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::result  [private]
```

Definition at line 153 of file setdiff.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

6.808 Rcpp::internal::UnwindData Struct Reference

```
#include <unwindProtect.h>
```

Public Attributes

- std::jmp_buf jmpbuf

6.808.1 Detailed Description

Definition at line 34 of file unwindProtect.h.

6.808.2 Member Data Documentation

6.808.2.1 jmpbuf

```
std::jmp_buf Rcpp::internal::UnwindData::jmpbuf
```

Definition at line 35 of file unwindProtect.h.

Referenced by Rcpp::internal::maybeJump(), and Rcpp::unwindProtect().

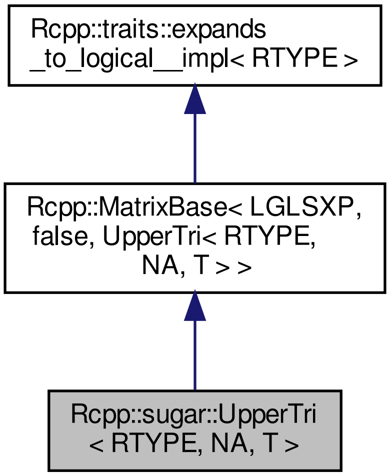
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/unwindProtect.h

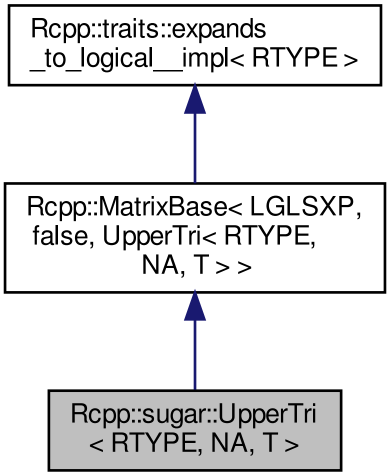
6.809 Rcpp::sugar::UpperTri< RTYPE, NA, T > Class Template Reference

```
#include <upper_tri.h>
```

Inheritance diagram for Rcpp::sugar::UpperTri< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::MatrixBase< LGLSXP, false, UpperTri< RTYPE, NA, T > >
Rcpp::sugar::UpperTri < RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::UpperTri< RTYPE, NA, T >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::MatrixBase< LGLSXP, false, UpperTri< RTYPE, NA, T > >
Rcpp::sugar::UpperTri < RTYPE, NA, T >

Public Types

- typedef Rcpp::MatrixBase< RTYPE, NA, T > MatBase

Public Member Functions

- UpperTri (const T &lhs, bool diag)
- int operator() (int i, int j) const
- R_xlen_t size () const
- int nrow () const
- int ncol () const

Private Types

- typedef bool(UpperTri::∗ Method) (int, int) const

Private Member Functions

- bool get_diag_true (int i, int j) const
- bool get_diag_false (int i, int j) const
- bool get (int i, int j) const

Private Attributes

- int nr
- int nc
- Method getter

Additional Inherited Members

6.809.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::UpperTri< RTYPE, NA, T >

Definition at line 30 of file upper_tri.h.

6.809.2 Member Typedef Documentation

6.809.2.1 MatBase

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::MatrixBase<RTYPE, NA, T> Rcpp::sugar::UpperTri< RTYPE, NA, T >::MatBase
```

Definition at line 32 of file upper_tri.h.

6.809.2.2 Method

```
template<int RTYPE, bool NA, typename T >
typedef bool(UpperTri::* Rcpp::sugar::UpperTri< RTYPE, NA, T >::Method) (int, int) const  [private]
```

Definition at line 47 of file upper_tri.h.

6.809.3 Constructor & Destructor Documentation

6.809.3.1 UpperTri()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UpperTri< RTYPE, NA, T >::UpperTri (
             const T & lhs,
             bool diag )  [inline]
```

Definition at line 34 of file upper_tri.h.

6.809.4 Member Function Documentation

6.809.4.1 get()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 56 of file upper_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, NA, T >::operator()().

6.809.4.2 get_diag_false()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get_diag_false (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 54 of file upper_tri.h.

6.809.4.3 get_diag_true()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get_diag_true (
             int i,
             int j ) const  [inline], [private]
```

Definition at line 52 of file upper_tri.h.

6.809.4.4 ncol()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::ncol ( ) const  [inline]
```

Definition at line 44 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc.

6.809.4.5 nrow()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nrow ( ) const  [inline]
```

Definition at line 43 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr.

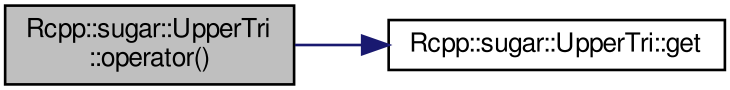
6.809.4.6 operator()()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::operator() (
             int i,
             int j ) const  [inline]
```

Definition at line 40 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::get().

Here is the call graph for this function:



6.809.4.7 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::UpperTri< RTYPE, NA, T >::size ( ) const  [inline]
```

Definition at line 42 of file upper_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc, and Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr.

6.809.5 Member Data Documentation

6.809.5.1 getter

```
template<int RTYPE, bool NA, typename T >
Method Rcpp::sugar::UpperTri< RTYPE, NA, T >::getter  [private]
```

Definition at line 50 of file upper_tri.h.

6.809.5.2 nc

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc  [private]
```

Definition at line 49 of file upper_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, NA, T >::ncol(), and Rcpp::sugar::UpperTri< RTYPE, NA, T >::size().

6.809.5.3 nr

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr  [private]
```

Definition at line 49 of file upper_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, NA, T >::nrow(), and Rcpp::sugar::UpperTri< RTYPE, NA, T >::size().

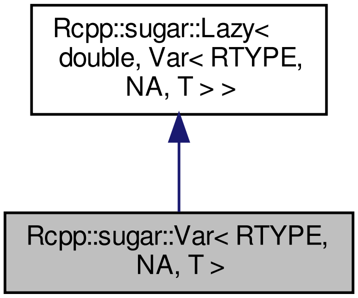
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/upper_tri.h

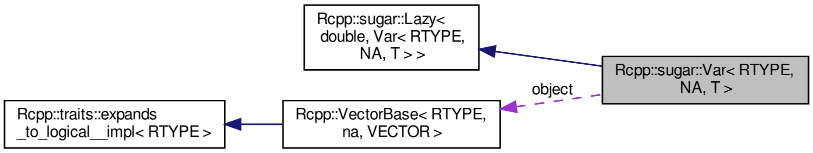
6.810 Rcpp::sugar::Var< RTYPE, NA, T > Class Template Reference

```
#include <var.h>
```

Inheritance diagram for Rcpp::sugar::Var< RTYPE, NA, T >:


Rcpp::sugar::Lazy< double, Var< RTYPE, NA, T > >
Rcpp::sugar::Var< RTYPE, NA, T >

Collaboration diagram for Rcpp::sugar::Var< RTYPE, NA, T >:


Rcpp::sugar::Lazy< double, Var< RTYPE, NA, T > >
Rcpp::sugar::Var< RTYPE, NA, T >
object
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE

Public Member Functions

- Var (const VEC_TYPE &object_)
- double get () const

Private Attributes

- const VEC_TYPE & object

6.810.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Var< RTYPE, NA, T >

Definition at line 30 of file var.h.

6.810.2 Member Typedef Documentation

6.810.2.1 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Var< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 32 of file var.h.

6.810.3 Constructor & Destructor Documentation

6.810.3.1 Var()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Var< RTYPE, NA, T >::Var (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 34 of file var.h.

6.810.4 Member Function Documentation

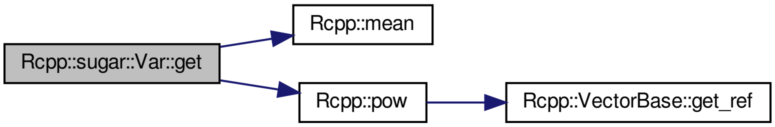
6.810.4.1 get()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::sugar::Var< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 36 of file var.h.

References Rcpp::mean(), and Rcpp::pow().

Here is the call graph for this function:



6.810.5 Member Data Documentation

6.810.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Var< RTYPE, NA, T >::object  [private]
```

Definition at line 46 of file var.h.

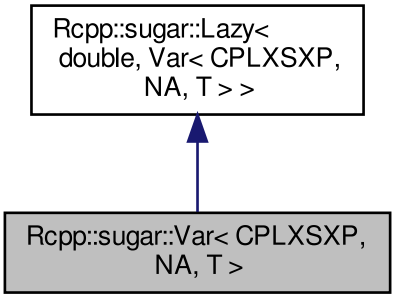
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/var.h

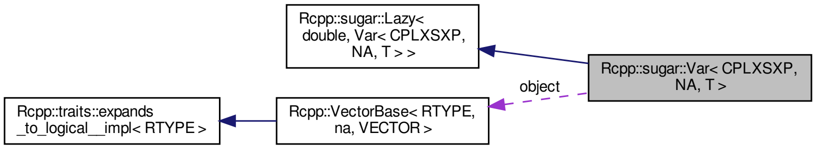
6.811 Rcpp::sugar::Var< CPLXSXP, NA, T > Class Template Reference

```
#include <var.h>
```

Inheritance diagram for Rcpp::sugar::Var< CPLXSXP, NA, T >:


Rcpp::sugar::Lazy< double, Var< CPLXSXP, NA, T > >
Rcpp::sugar::Var< CPLXSXP, NA, T >

Collaboration diagram for Rcpp::sugar::Var< CPLXSXP, NA, T >:


Rcpp::sugar::Lazy< double, Var< CPLXSXP, NA, T > >
object
Rcpp::sugar::Var< CPLXSXP, NA, T >
Rcpp::traits::expands _to_logical__impl< RTYPE >
Rcpp::VectorBase< RTYPE, na, VECTOR >

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

6.811.1 Detailed Description

template<bool NA, typename T>
class Rcpp::sugar::Var< CPLXSXP, NA, T >

Definition at line 50 of file var.h.

6.811.2 Member Typedef Documentation

6.811.2.1 VEC_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::Var< CPLXSXP, NA, T >::VEC_TYPE
```

Definition at line 52 of file var.h.

6.811.3 Constructor & Destructor Documentation

6.811.3.1 Var()

```
template<bool NA, typename T >
Rcpp::sugar::Var< CPLXSXP, NA, T >::Var (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 54 of file var.h.

6.811.4 Member Function Documentation

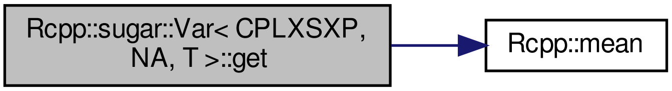
6.811.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Var< CPLXSXP, NA, T >::get ( ) const  [inline]
```

Definition at line 56 of file var.h.

References Rcpp::mean().

Here is the call graph for this function:



6.811.5 Member Data Documentation

6.811.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Var< CPLXSXP, NA, T >::object  [private]
```

Definition at line 68 of file var.h.

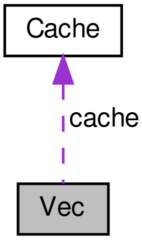
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/var.h

6.812 Vec Class Reference

```
#include <convolve10_cpp.h>
```

Collaboration diagram for Vec:



Public Types

- typedef double & proxy
- typedef double & proxy

Public Member Functions

- Vec (double ∗data_)
- proxy operator[ ] (int i)
- proxy operator[ ] (int i) const
- Vec (double ∗data_)
- double & operator[ ] (int i)
- Vec (double ∗data_)
- proxy operator[ ] (int i)
- proxy operator[ ] (int i) const

Private Attributes

- Cache cache
- double ∗ data

6.812.1 Detailed Description

Definition at line 16 of file convolve10_cpp.h.

6.812.2 Member Typedef Documentation

6.812.2.1 proxy [1/2]

```
typedef double& Vec::proxy
```

Definition at line 18 of file convolve10_cpp.h.

6.812.2.2 proxy [2/2]

```
typedef double& Vec::proxy
```

Definition at line 31 of file convolve9_cpp.cpp.

6.812.3 Constructor & Destructor Documentation

6.812.3.1 Vec() [1/3]

```
Vec::Vec (
            double * data_ )  [inline]
```

Definition at line 20 of file convolve10_cpp.h.

6.812.3.2 Vec() [2/3]

```
Vec::Vec (
            double * data_ )  [inline]
```

Definition at line 16 of file convolve8_cpp.cpp.

6.812.3.3 Vec() [3/3]

```
Vec::Vec (
             double * data_ )  [inline]
```

Definition at line 33 of file convolve9_cpp.cpp.

6.812.4 Member Function Documentation

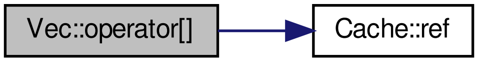
6.812.4.1 operator[]() [1/5]

```
proxy Vec::operator[] (
             int i )  [inline]
```

Definition at line 21 of file convolve10_cpp.h.

References cache, and Cache::ref().

Here is the call graph for this function:



6.812.4.2 operator[]() [2/5]

```
double& Vec::operator[] (
             int i )  [inline]
```

Definition at line 17 of file convolve8_cpp.cpp.

References data.

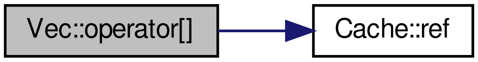
6.812.4.3 operator[]() [3/5]

```
proxy Vec::operator[] (
             int i )  [inline]
```

Definition at line 34 of file convolve9_cpp.cpp.

References cache, and Cache::ref().

Here is the call graph for this function:



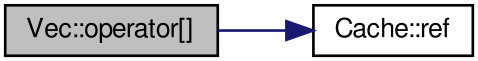
6.812.4.4 operator[]() [4/5]

```
proxy Vec::operator[] (
             int i ) const  [inline]
```

Definition at line 22 of file convolve10_cpp.h.

References cache, and Cache::ref().

Here is the call graph for this function:



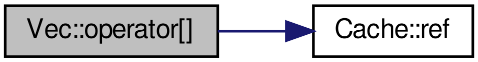
6.812.4.5 operator[]() [5/5]

```
proxy Vec::operator[] (
             int i ) const  [inline]
```

Definition at line 35 of file convolve9_cpp.cpp.

References cache, and Cache::ref().

Here is the call graph for this function:



6.812.5 Member Data Documentation

6.812.5.1 cache

```
Cache Vec::cache  [private]
```

Definition at line 25 of file convolve10_cpp.h.

Referenced by operator[ ]().

6.812.5.2 data

```
double* Vec::data  [private]
```

Definition at line 20 of file convolve8_cpp.cpp.

Referenced by operator[ ]().

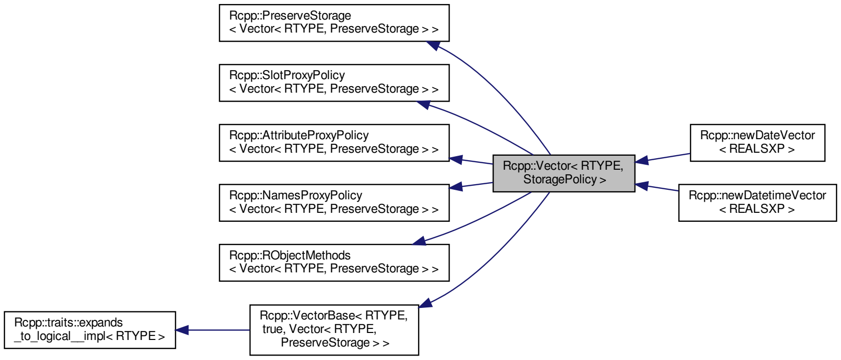
The documentation for this class was generated from the following files:

- inst/examples/ConvolveBenchmarks/convolve10_cpp.h
- inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
- inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp

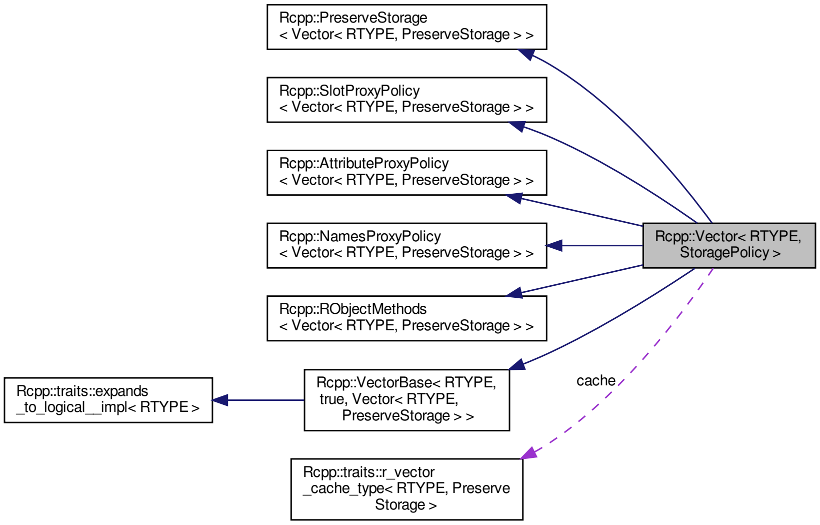
6.813 Rcpp::Vector< RTYPE, StoragePolicy > Class Template Reference

```
#include <Vector.h>
```

Inheritance diagram for Rcpp::Vector< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::Vector< RTYPE, StoragePolicy >
Rcpp::newDateVector
< REALSXP >
Rcpp::newDatetimeVector
< REALSXP >

Collaboration diagram for Rcpp::Vector< RTYPE, StoragePolicy >:


Rcpp::PreserveStorage
< Vector< RTYPE, PreserveStorage > >
Rcpp::SlotProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::AttributeProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage > >
Rcpp::RObjectMethods
< Vector< RTYPE, PreserveStorage > >
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > >
Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::Vector< RTYPE, StoragePolicy >
cache
Rcpp::traits::r_vector_cache_type< RTYPE, PreserveStorage >

Public Types

- typedef StoragePolicy< Vector > Storage
- typedef traits::r_vector_proxy< RTYPE, StoragePolicy >::type Proxy
- typedef traits::r_vector_const_proxy< RTYPE, StoragePolicy >::type const_Proxy

- typedef traits::r_vector_name_proxy< RTYPE, StoragePolicy >::type NameProxy
- typedef traits::r_vector_proxy< RTYPE, StoragePolicy >::type value_type
- typedef traits::r_vector_iterator< RTYPE, StoragePolicy >::type iterator
- typedef traits::r_vector_const_iterator< RTYPE, StoragePolicy >::type const_iterator
- typedef traits::init_type< RTYPE >::type init_type
- typedef traits::r_vector_element_converter< RTYPE >::type converter_type
- typedef traits::storage_type< RTYPE >::type stored_type
- typedef internal::RangeIndexer< RTYPE, true, Vector > Indexer

Public Member Functions

- Vector ()
- Vector (const Vector &other)
- Vector & operator= (const Vector &rhs)
- Vector (SEXP x)
- template<typename Proxy >
  Vector (const GenericProxy< Proxy > &proxy)
- Vector (const no_init_vector &obj)
- template<typename T >
  Vector (const T &size, const stored_type &u, typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type ∗=0)
- Vector (const int &size, const stored_type &u)
- Vector (const std::string &st)
- Vector (const char ∗st)
- template<typename T >
  Vector (const T &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
- const_iterator cbegin () const
- const_iterator cend () const
- Proxy operator[ ] (R_xlen_t i)
- const_Proxy operator[ ] (R_xlen_t i) const
- Proxy operator() (const size_t &i)
- const_Proxy operator() (const size_t &i) const
- Proxy at (const size_t &i)
- const_Proxy at (const size_t &i) const
- Proxy operator() (const size_t &i, const size_t &j)
- const_Proxy operator() (const size_t &i, const size_t &j) const
- NameProxy operator[ ] (const std::string &name)
- NameProxy operator() (const std::string &name)
- NameProxy operator[ ] (const std::string &name) const
- NameProxy operator() (const std::string &name) const
- operator RObject () const
- template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
  SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > operator[ ] (const VectorBase< RHS↩ _RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
  const SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > operator[ ] (const VectorBase< RHS_RTYPE, RHS_NA, RHS_T > &rhs) const
- Vector & sort (bool decreasing=false)
- template<typename InputIterator >
  void assign (InputIterator first, InputIterator last)
- template<typename T >
  void push_back (const T &object)

- template<typename T >
  void push_back (const T &object, const std::string &name)
- template<typename T >
  void push_front (const T &object)
- template<typename T >
  void push_front (const T &object, const std::string &name)
- template<typename T >
  iterator insert (iterator position, const T &object)
- template<typename T >
  iterator insert (int position, const T &object)
- iterator erase (int position)
- iterator erase (iterator position)
- iterator erase (int first, int last)
- iterator erase (iterator first, iterator last)
- void update (SEXP)
- Indexer operator[ ] (const Range &range)
- template<typename EXPR_VEC >
  Vector & operator+= (const VectorBase< RTYPE, true, EXPR_VEC > &rhs)
- template<typename EXPR_VEC >
  Vector & operator+= (const VectorBase< RTYPE, false, EXPR_VEC > &rhs)
- bool containsElementNamed (const char ∗target) const
- int findName (const std::string &name) const
- SEXP eval () const
- SEXP eval (SEXP env) const

Static Public Member Functions

- static stored_type get_na ()
- static bool is_na (stored_type x)
- template<typename InputIterator >
  static Vector import (InputIterator first, InputIterator last)
- template<typename InputIterator , typename F >
  static Vector import_transform (InputIterator first, InputIterator last, F f)
- template<typename U >
  static void replace_element (iterator it, SEXP names, R_xlen_t index, const U &u)
- template<typename U >
  static void replace_element__dispatch (traits::false_type, iterator it, SEXP names, R_xlen_t index, const U &u)
- template<typename U >
  static void replace_element__dispatch (traits::true_type, iterator it, SEXP names, R_xlen_t index, const U &u)
- template<typename U >
  static void replace_element__dispatch__isArgument (traits::false_type, iterator it, SEXP names, R_xlen_t index, const U &u)
- template<typename U >
  static void replace_element__dispatch__isArgument (traits::true_type, iterator it, SEXP names, R_xlen_t index, const U &u)
- static Vector create ()

Public Attributes

- traits::r_vector_cache_type< RTYPE, StoragePolicy >::type cache

Protected Member Functions

- int ∗ dims () const
- void init ()

Private Member Functions

- void push_back__impl (const stored_type &object, traits::true_type)
- void push_back__impl (const stored_type &object, traits::false_type)
- void push_back_name__impl (const stored_type &object, const std::string &name, traits::true_type)
- void push_back_name__impl (const stored_type &object, const std::string &name, traits::false_type)
- void push_front__impl (const stored_type &object, traits::true_type)
- void push_front__impl (const stored_type &object, traits::false_type)
- void push_front_name__impl (const stored_type &object, const std::string &name, traits::true_type)
- void push_front_name__impl (const stored_type &object, const std::string &name, traits::false_type)
- iterator insert__impl (iterator position, const stored_type &object_, traits::true_type)
- iterator insert__impl (iterator position, const stored_type &object, traits::false_type)
- iterator erase_single__impl (iterator position)
- iterator erase_range__impl (iterator first, iterator last)
- template<typename T >
  void assign_sugar_expression (const T &x)
- template<typename T >
  void assign_object (const T &x, traits::true_type)
- template<typename T >
  void assign_object (const T &x, traits::false_type)
- template<bool NA, typename VEC >
  void import_sugar_expression (const Rcpp::VectorBase< RTYPE, NA, VEC > &other, traits::false_type)
- template<bool NA, typename VEC >
  void import_sugar_expression (const Rcpp::VectorBase< RTYPE, NA, VEC > &other, traits::true_type)
- template<typename T >
  void import_expression (const T &other, R_xlen_t n)
- template<typename T >
  void fill_or_generate (const T &t)
- template<typename T >
  void fill_or_generate__impl (const T &gen, traits::true_type)
- template<typename T >
  void fill_or_generate__impl (const T &t, traits::false_type)
- template<typename U >
  void fill__dispatch (traits::false_type, const U &u)
- template<typename U >
  void fill__dispatch (traits::true_type, const U &u)

6.813.1 Detailed Description

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::Vector< RTYPE, StoragePolicy >

Examples

ConvolveBenchmarks/convolve3_cpp.cpp, and ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 29 of file Vector.h.

6.813.2 Member Typedef Documentation

6.813.2.1 const_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_const_iterator<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, Storage↩
Policy >::const_iterator
```

Definition at line 47 of file Vector.h.

6.813.2.2 const_Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_const_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, Storage↩
Policy >::const_Proxy
```

Definition at line 43 of file Vector.h.

6.813.2.3 converter_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_element_converter<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >↩
::converter_type
```

Definition at line 49 of file Vector.h.

6.813.2.4 Indexer

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef internal::RangeIndexer<RTYPE,true,Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Indexer
```

Definition at line 545 of file Vector.h.

6.813.2.5 init_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::init_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::init_type
```

Definition at line 48 of file Vector.h.

6.813.2.6 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_iterator<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::iterator
```

Definition at line 46 of file Vector.h.

6.813.2.7 NameProxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_name_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, Storage↩
Policy >::NameProxy
```

Definition at line 44 of file Vector.h.

6.813.2.8 Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::Proxy
```

Definition at line 42 of file Vector.h.

6.813.2.9 Storage

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef StoragePolicy<Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Storage
```

Definition at line 39 of file Vector.h.

6.813.2.10 stored_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::storage_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::stored_type
```

Definition at line 50 of file Vector.h.

6.813.2.11 value_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::value_type
```

Definition at line 45 of file Vector.h.

6.813.3 Constructor & Destructor Documentation

6.813.3.1 Vector() [1/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector ( ) [inline]
```

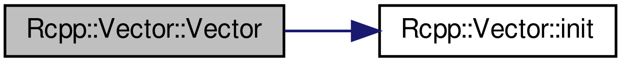
Default constructor. Creates a vector of the appropriate type and 0 length

Definition at line 56 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::create(), and Rcpp::Vector< RTYPE, StoragePolicy >↩ ::import_transform().

Here is the call graph for this function:



6.813.3.2 Vector() [2/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const Vector< RTYPE, StoragePolicy > & other )  [inline]
```

copy constructor. shallow copy of the SEXP

Definition at line 64 of file Vector.h.

6.813.3.3 Vector() [3/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             SEXP x )  [inline]
```

Definition at line 72 of file Vector.h.

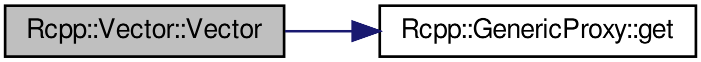
6.813.3.4 Vector() [4/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename Proxy >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const GenericProxy< Proxy > & proxy )  [inline]
```

Definition at line 78 of file Vector.h.

References Rcpp::GenericProxy< Proxy >::get().

Here is the call graph for this function:



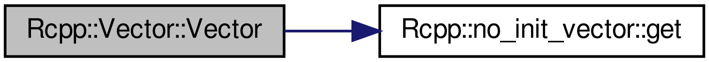
6.813.3.5 Vector() [5/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
            const no_init_vector & obj )  [inline], [explicit]
```

Definition at line 83 of file Vector.h.

References Rcpp::no_init_vector::get().

Here is the call graph for this function:



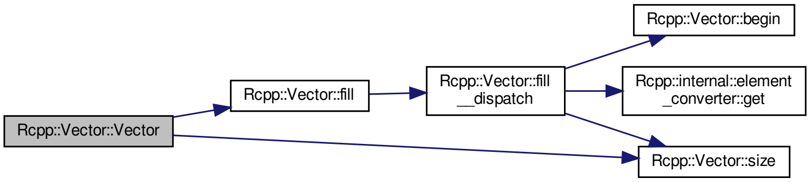
6.813.3.6 Vector() [6/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
            const T & size,
            const stored_type & u,
            typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 88 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill(), RCPP_DEBUG_2, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



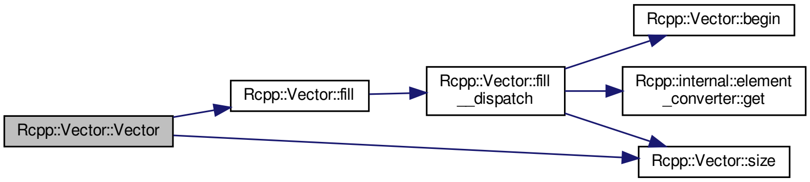
6.813.3.7 Vector() [7/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
            const int & size,
            const stored_type & u ) [inline]
```

Definition at line 95 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill(), RCPP_DEBUG_2, and Rcpp::Vector< RTYPE, Storage↩ Policy >::size().

Here is the call graph for this function:



6.813.3.8 Vector() [8/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
            const std::string & st ) [inline]
```

Definition at line 102 of file Vector.h.

References RCPP_DEBUG_2.

6.813.3.9 Vector() [9/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
            const char * st ) [inline]
```

Definition at line 108 of file Vector.h.

References RCPP_DEBUG_2.

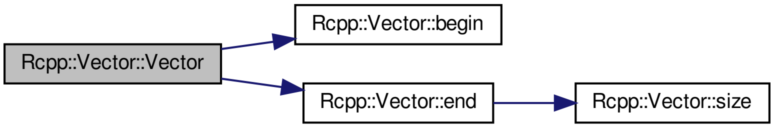
6.813.3.10 Vector() [10/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const T & siz,
             stored_type(*)(void) gen,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 114 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and RCPP_DEBUG_2.

Here is the call graph for this function:



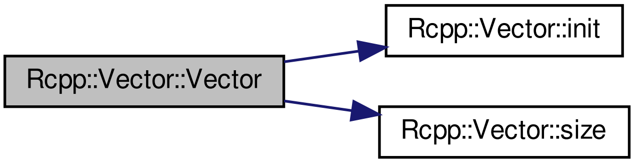
6.813.3.11 Vector() [11/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             T size,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 123 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



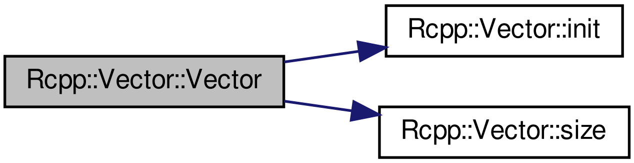
6.813.3.12 Vector() [12/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const int & size )  [inline]
```

Definition at line 129 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



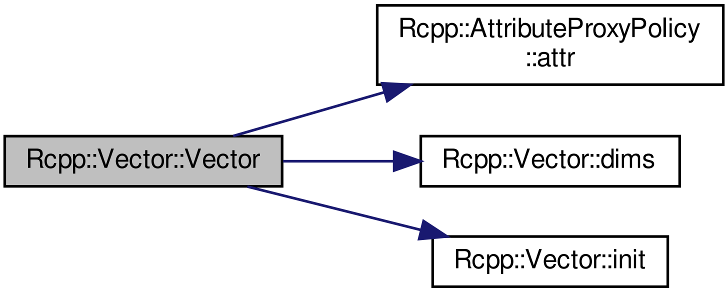
6.813.3.13 Vector() [13/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const Dimension & dims )  [inline]
```

Definition at line 134 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::dims(), and Rcpp::Vector< RTYPE, StoragePolicy >::init().

Here is the call graph for this function:



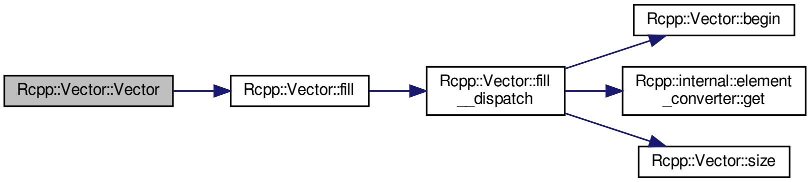
6.813.3.14 Vector() [14/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             T value,
             typename Rcpp::traits::enable_if< traits::is_bool< T >::value &&RTYPE==LGLSXP, void
>::type * = 0 ) [inline]
```

Definition at line 146 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill().

Here is the call graph for this function:



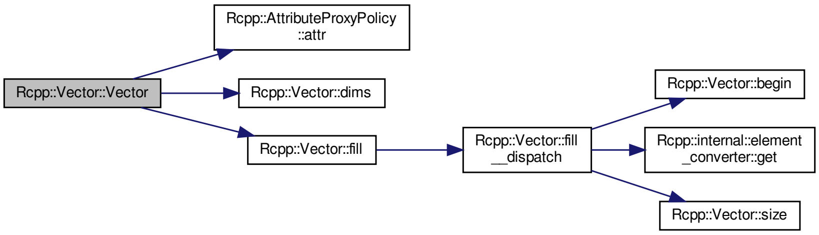
6.813.3.15 Vector() [15/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const Dimension & dims,
             const U & u ) [inline]
```

Definition at line 153 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::dims(), Rcpp::↩ Vector< RTYPE, StoragePolicy >::fill(), and RCPP_DEBUG_2.

Here is the call graph for this function:



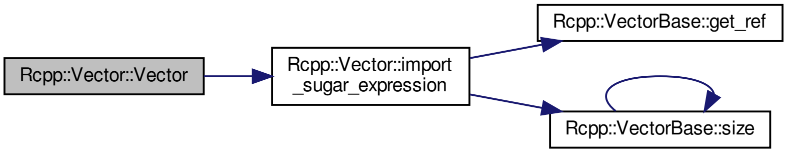
6.813.3.16 Vector() [16/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const VectorBase< RTYPE, NA, VEC > & other ) [inline]
```

Definition at line 163 of file Vector.h.

References DEMANGLE, Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression(), and RCPP_DEBUG_2.

Here is the call graph for this function:



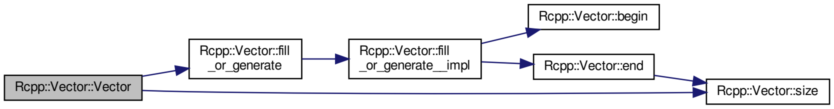
6.813.3.17 Vector() [17/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const T & size,
             const U & u,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 169 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate(), RCPP_DEBUG_2, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.3.18 Vector() [18/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const sugar::SingleLogicalResult< NA, T > & obj )  [inline]
```

Definition at line 177 of file Vector.h.

References DEMANGLE, and RCPP_DEBUG_2.

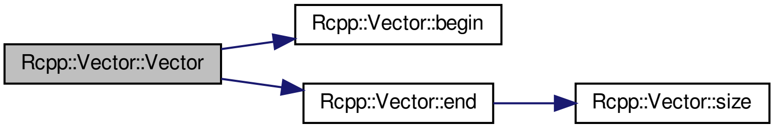
6.813.3.19 Vector() [19/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const T & siz,
             stored_type(*)(U1) gen,
             const U1 & u1,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 184 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and RCPP_DEBUG_2.

Here is the call graph for this function:



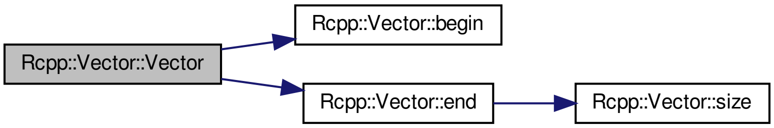
6.813.3.20 Vector() [20/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 , typename U2 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const T & siz,
             stored_type(*)(U1, U2) gen,
             const U1 & u1,
             const U2 & u2,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 193 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and RCPP_DEBUG_2.

Here is the call graph for this function:



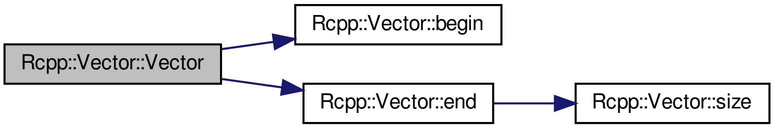
6.813.3.21 Vector() [21/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 , typename U2 , typename U3 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             const T & siz,
             stored_type(*)(U1, U2, U3) gen,
             const U1 & u1,
             const U2 & u2,
             const U3 & u3,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 202 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and RCPP_DEBUG_2.

Here is the call graph for this function:



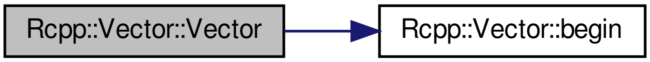
6.813.3.22 Vector() [22/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             InputIterator first,
             InputIterator last )  [inline]
```

Definition at line 211 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP_DEBUG_1.

Here is the call graph for this function:



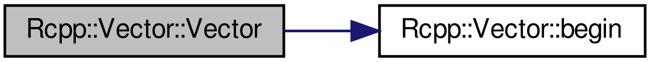
6.813.3.23 Vector() [23/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             InputIterator first,
             InputIterator last,
             T n,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 )  [inline]
```

Definition at line 218 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP_DEBUG_2.

Here is the call graph for this function:



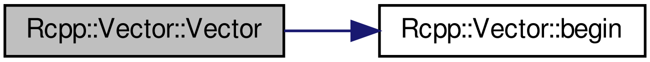
6.813.3.24 Vector() [24/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename Func >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             InputIterator first,
             InputIterator last,
             Func func ) [inline]
```

Definition at line 226 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP_DEBUG_1.

Here is the call graph for this function:



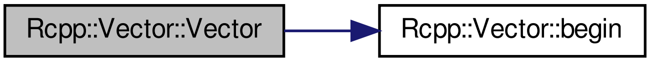
6.813.3.25 Vector() [25/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename Func , typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
             InputIterator first,
             InputIterator last,
             Func func,
             T n,
             typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]
```

Definition at line 233 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP_DEBUG_2.

Here is the call graph for this function:



6.813.4 Member Function Documentation

6.813.4.1 assign()

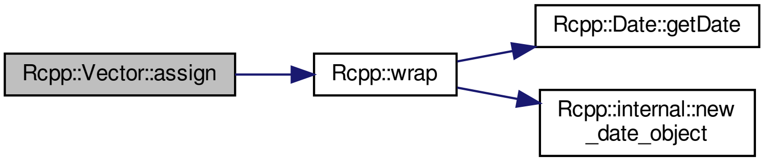
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign (
             InputIterator first,
             InputIterator last )  [inline]
```

Definition at line 430 of file Vector.h.

References Rcpp::wrap().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::import().

Here is the call graph for this function:



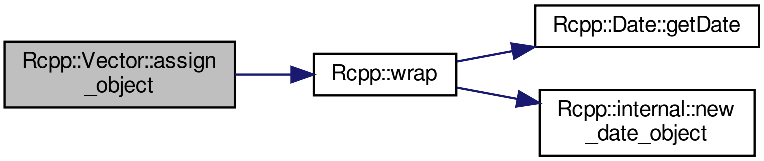
6.813.4.2 assign_object() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_object (
             const T & x,
             traits::false_type  )  [inline], [private]
```

Definition at line 1059 of file Vector.h.

References Rcpp::wrap().

Here is the call graph for this function:



6.813.4.3 assign_object() [2/2]

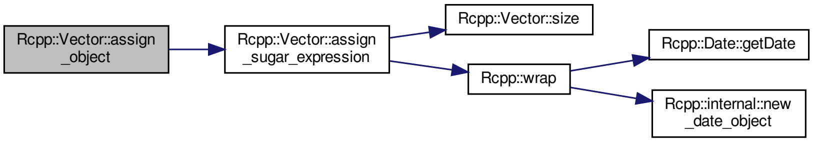
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_object (
            const T & x,
            traits::true_type ) [inline], [private]
```

Definition at line 1053 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



6.813.4.4 assign_sugar_expression()

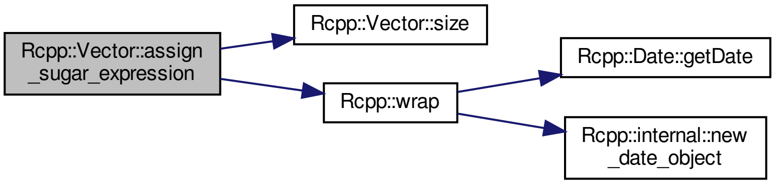
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression (
             const T & x )  [inline], [private]
```

Definition at line 1038 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::wrap().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::assign_object().

Here is the call graph for this function:



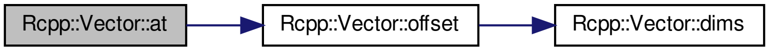
6.813.4.5 at() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (
             const size_t & i )  [inline]
```

Definition at line 351 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



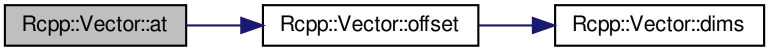
6.813.4.6 at() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (
            const size_t & i ) const  [inline]
```

Definition at line 354 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



6.813.4.7 begin() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( )  [inline]
```

Examples

ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 334 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

Referenced by Rcpp::ListOf< T >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::cbegin(), compileAttributes(), convolve10cpp(), convolve12cpp(), convolve4cpp(), convolve8cpp(), convolve9cpp(), Rcpp::exception::copy_stack_trace_to_r(), Rcpp::sugar::EmpiricalSample(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl(), fastLm(), Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch(), Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::import_expression(), Rcpp::Matrix< RTYPE, StoragePolicy >::import_matrix_expression(), Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch(), Rcpp::sugar::Table< RTYPE, TABLE_T >::Table(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

6.813.4.8 begin() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( ) const  [inline]
```

Definition at line 336 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

6.813.4.9 cbegin()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::cbegin ( ) const  [inline]
```

Definition at line 338 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

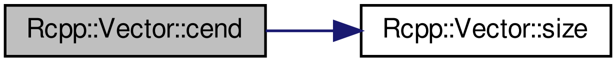
6.813.4.10 cend()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::cend ( ) const  [inline]
```

Definition at line 339 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.4.11 containsElementNamed()

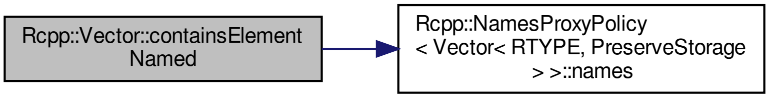
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
bool Rcpp::Vector< RTYPE, StoragePolicy >::containsElementNamed (
            const char * target ) const  [inline]
```

Does this vector have an element with the target name

Definition at line 586 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and RCPP_GET_NAMES.

Here is the call graph for this function:


Rcpp::Vector::containsElement Named
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.12 create()

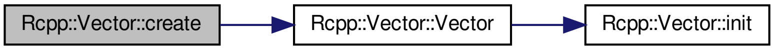
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::create ( )  [inline], [static]
```

Definition at line 1122 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >(), Rcpp::shush_about_NA(), and Rcpp::trimws().

Here is the call graph for this function:



6.813.4.13 dims()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int* Rcpp::Vector< RTYPE, StoragePolicy >::dims ( ) const  [inline], [protected]
```

Definition at line 613 of file Vector.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::cols(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

6.813.4.14 end() [1/2]

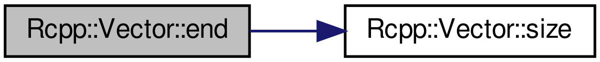
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( )  [inline]
```

Definition at line 335 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::cend(), compileAttributes(), Rcpp::sugar::EmpiricalSample(), Rcpp::ListOf< T >::end(), Rcpp::Matrix< RTYPE, StoragePolicy >::end(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch(), Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl(), Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch(), Rcpp::sugar::Table< RTYPE, TABLE_T >::Table(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



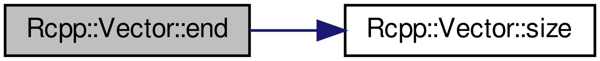
6.813.4.15 end() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( ) const  [inline]
```

Definition at line 337 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



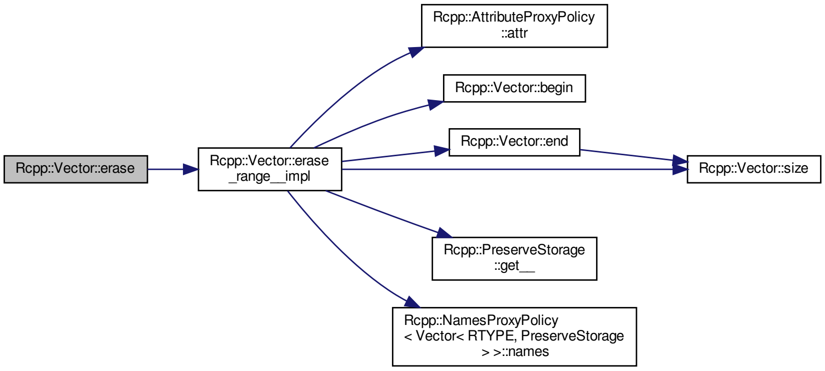
6.813.4.16 erase() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
             int first,
             int last )  [inline]
```

Definition at line 500 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::erase
Rcpp::Vector::erase _range__impl
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.17 erase() [2/4]

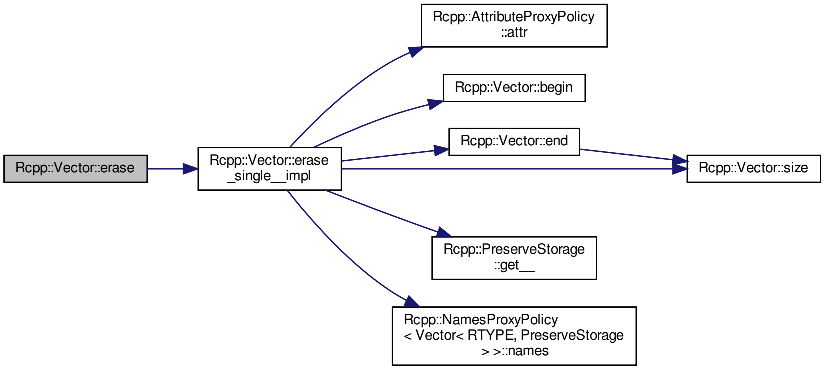
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
            int position ) [inline]
```

Definition at line 492 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl().

Referenced by Rcpp::DataFrame_Impl< StoragePolicy >::from_list().

Here is the call graph for this function:


Rcpp::Vector::erase
Rcpp::Vector::erase _single__impl
Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

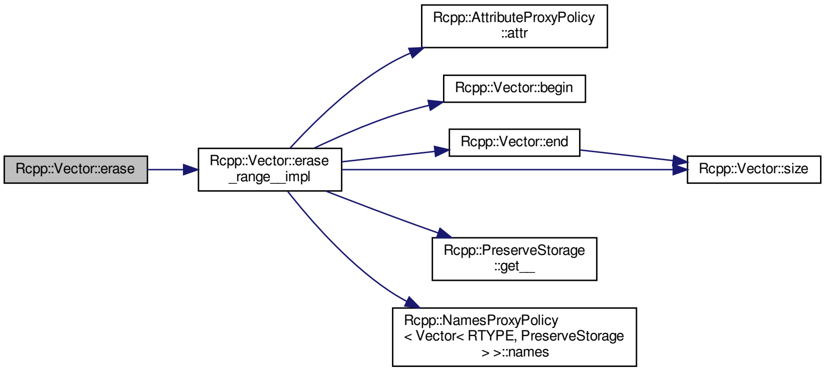
6.813.4.18 erase() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
            iterator first,
            iterator last ) [inline]
```

Definition at line 505 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl().

Here is the call graph for this function:


Rcpp::Vector::erase
Rcpp::Vector::erase_range__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

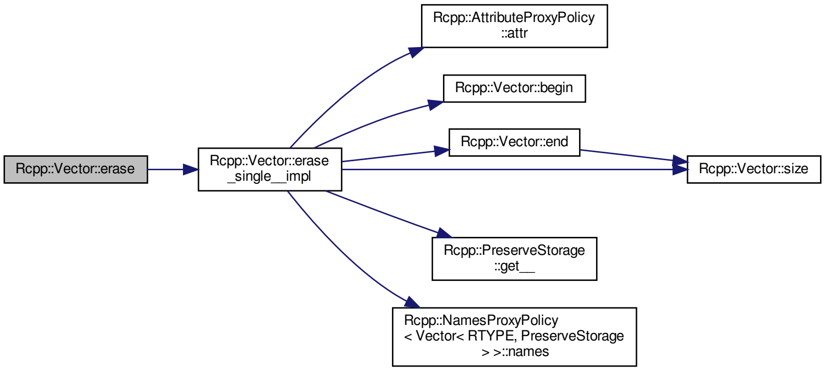
6.813.4.19 erase() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
             iterator position )  [inline]
```

Definition at line 496 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl().

Here is the call graph for this function:


Rcpp::Vector::erase
Rcpp::Vector::erase_single__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

6.813.4.20 erase_range__impl()

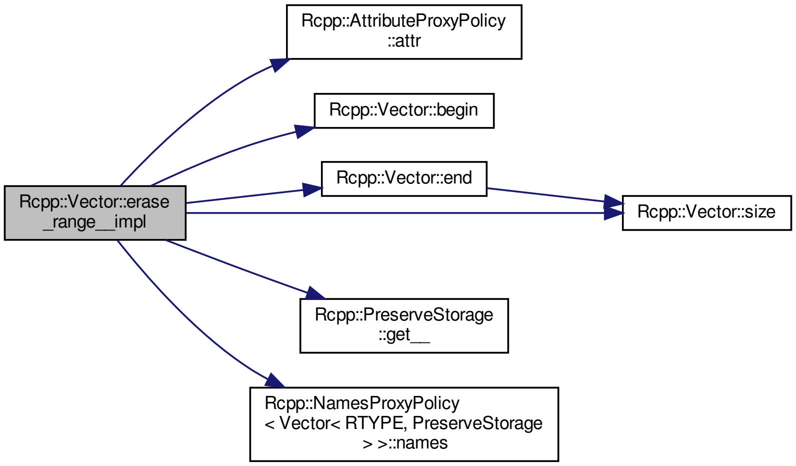
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl (
            iterator first,
            iterator last ) [inline], [private]
```

Definition at line 978 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::erase _range__impl
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.21 erase_single__impl()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl (
            iterator position ) [inline], [private]
```

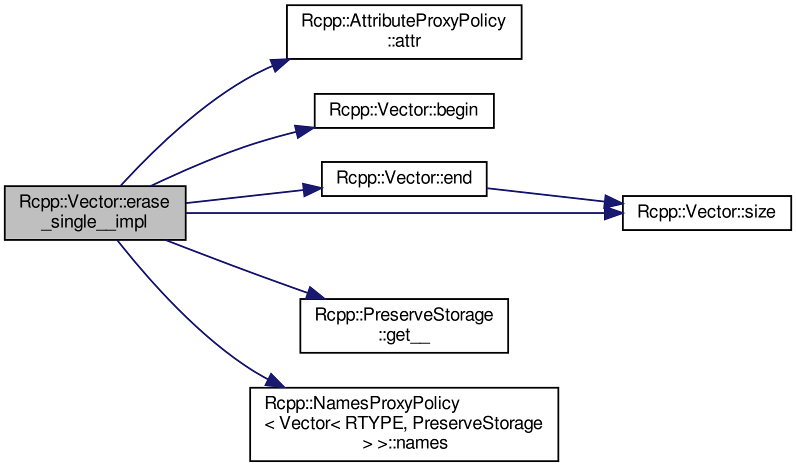
Definition at line 924 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy<

Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase().

Here is the call graph for this function:



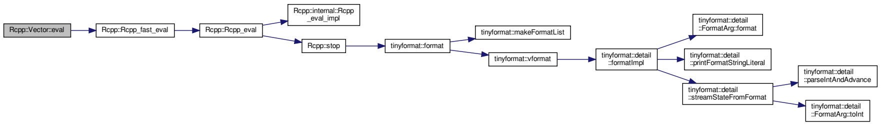
6.813.4.22 eval() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval ( ) const  [inline]
```

Definition at line 1130 of file Vector.h.

References Rcpp::Rcpp_fast_eval().

Here is the call graph for this function:



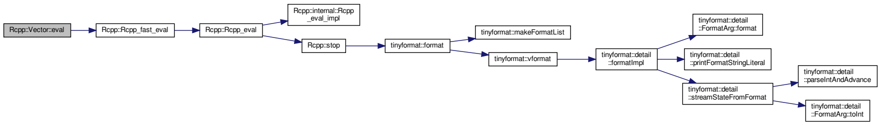
6.813.4.23 eval() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval (
             SEXP env ) const  [inline]
```

Definition at line 1134 of file Vector.h.

References Rcpp::Rcpp_fast_eval().

Here is the call graph for this function:


Rcpp::Vector::eval
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

6.813.4.24 fill()

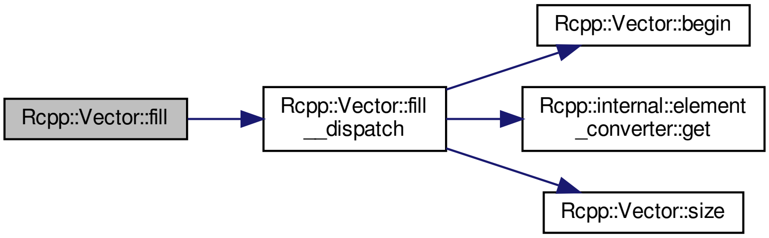
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill (
             const U & u )  [inline]
```

Definition at line 330 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:


Rcpp::Vector::fill
Rcpp::Vector::fill__dispatch
Rcpp::Vector::begin
Rcpp::internal::element_converter::get
Rcpp::Vector::size

6.813.4.25 fill__dispatch() [1/2]

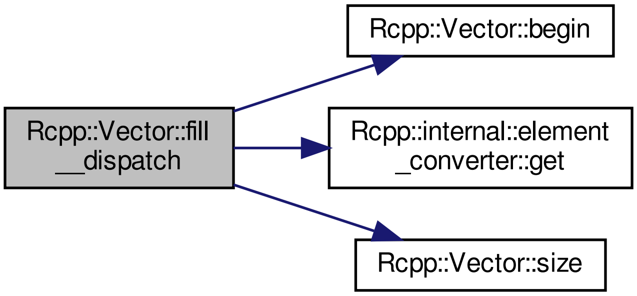
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch (
            traits::false_type ,
            const U & u ) [inline], [private]
```

Definition at line 1106 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill().

Here is the call graph for this function:



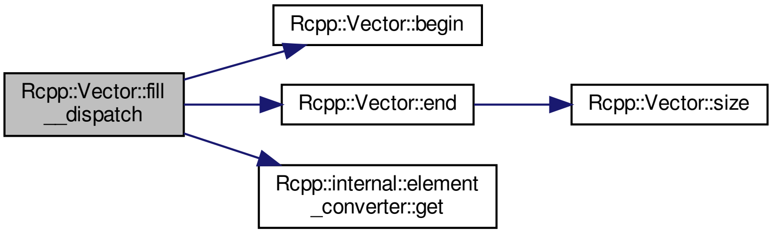
6.813.4.26 fill__dispatch() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch (
            traits::true_type ,
            const U & u ) [inline], [private]
```

Definition at line 1116 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::internal::element_converter< RTYPE >::get().

Here is the call graph for this function:



6.813.4.27 fill_or_generate()

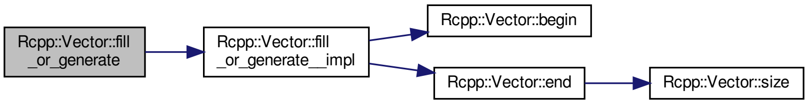
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate (
             const T & t )  [inline], [private]
```

Definition at line 1089 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



6.813.4.28 fill_or_generate__impl() [1/2]

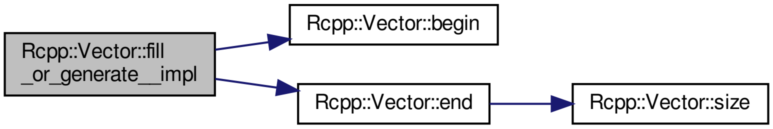
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl (
             const T & gen,
             traits::true_type )  [inline], [private]
```

Definition at line 1094 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::end().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate().

Here is the call graph for this function:



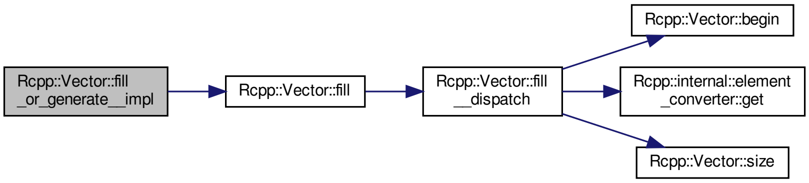
6.813.4.29 fill_or_generate__impl() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl (
             const T & t,
             traits::false_type )  [inline], [private]
```

Definition at line 1101 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill().

Here is the call graph for this function:



6.813.4.30 findName()

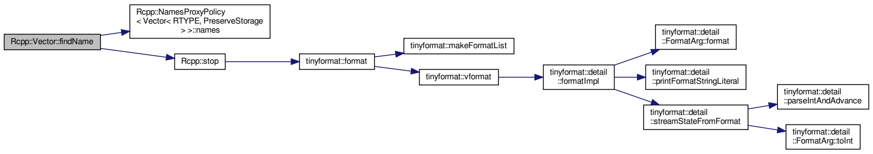
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Vector< RTYPE, StoragePolicy >::findName (
            const std::string & name ) const  [inline]
```

Definition at line 597 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::stop().

Referenced by Rcpp::ListOf< T >::operator[ ]().

Here is the call graph for this function:


Rcpp::Vector::findName
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

6.813.4.31 get_na()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static stored_type Rcpp::Vector< RTYPE, StoragePolicy >::get_na ( )  [inline], [static]
```

Definition at line 252 of file Vector.h.

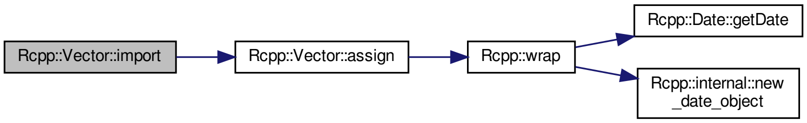
6.813.4.32 import()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::import (
            InputIterator first,
            InputIterator last )  [inline], [static]
```

Definition at line 440 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::assign().

Here is the call graph for this function:



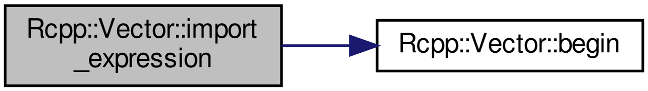
6.813.4.33 import_expression()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_expression (
             const T & other,
             R_xlen_t n )  [inline], [private]
```

Definition at line 1083 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP_LOOP_UNROLL.

Here is the call graph for this function:



6.813.4.34 import_sugar_expression() [1/2]

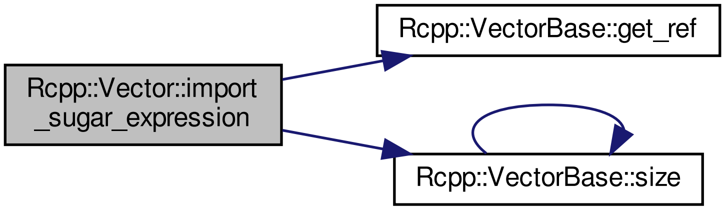
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression (
             const Rcpp::VectorBase< RTYPE, NA, VEC > & other,
             traits::false_type  )  [inline], [private]
```

Definition at line 1067 of file Vector.h.

References DEMANGLE, Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), Rcpp::NA, RCPP_DEBUG_4, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



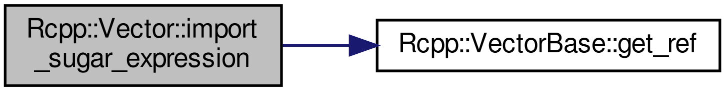
6.813.4.35 import_sugar_expression() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression (
             const Rcpp::VectorBase< RTYPE, NA, VEC > & other,
             traits::true_type ) [inline], [private]
```

Definition at line 1076 of file Vector.h.

References DEMANGLE, Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), Rcpp::NA, and RCPP_DEBUG_4.

Here is the call graph for this function:



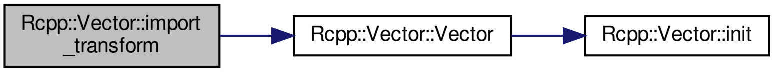
6.813.4.36 import_transform()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename F >
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::import_transform (
            InputIterator first,
            InputIterator last,
            F f ) [inline], [static]
```

Definition at line 447 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



6.813.4.37 init()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::init ( ) [inline], [protected]
```

Definition at line 617 of file Vector.h.

References RCPP_DEBUG_2.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Vector< RTYPE, StoragePolicy >::↩
Vector().

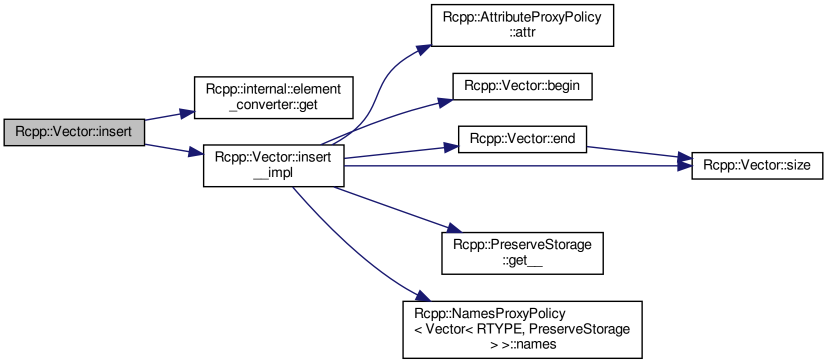
6.813.4.38 insert() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert (
            int position,
            const T & object )  [inline]
```

Definition at line 486 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl().

Here is the call graph for this function:


Rcpp::Vector::insert
Rcpp::internal::element_converter::get
Rcpp::Vector::insert__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

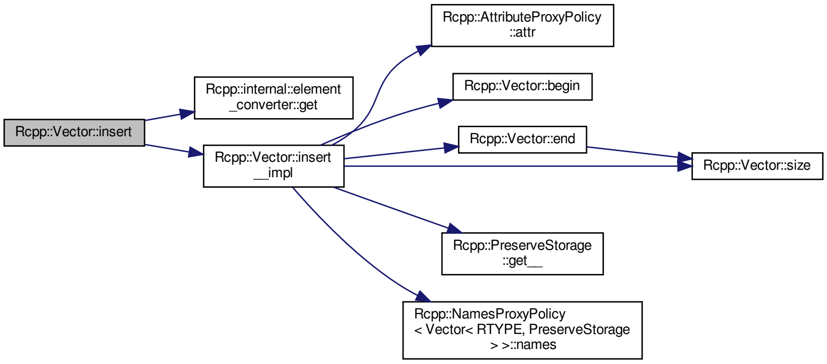
6.813.4.39 insert() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert (
            iterator position,
            const T & object )  [inline]
```

Definition at line 479 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >↩ ::insert__impl().

Here is the call graph for this function:


Rcpp::Vector::insert
Rcpp::internal::element_converter::get
Rcpp::Vector::insert__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

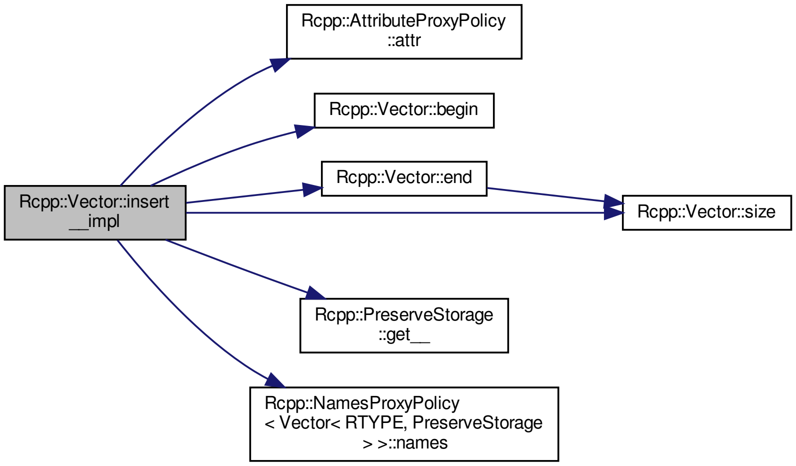
6.813.4.40 insert__impl() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl (
             iterator position,
             const stored_type & object,
             traits::false_type ) [inline], [private]
```

Definition at line 884 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::insert __impl
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.41 insert__impl() [2/2]

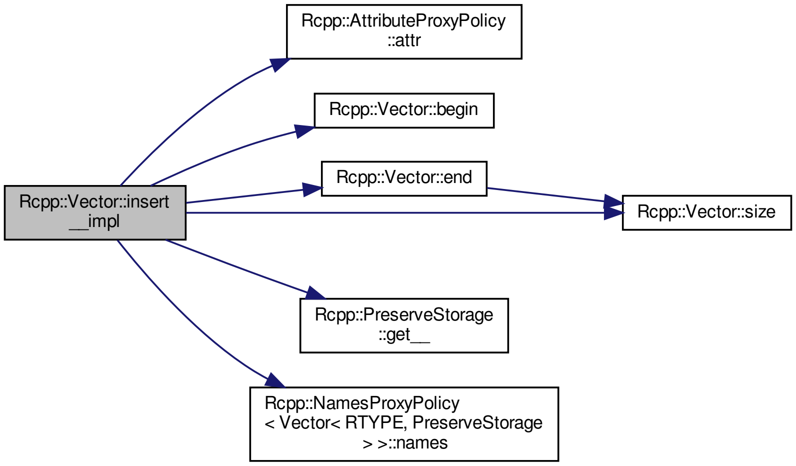
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl (
            iterator position,
            const stored_type & object_,
            traits::true_type ) [inline], [private]
```

Definition at line 843 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp↩ ::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::insert().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::insert __impl
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.42 is_na()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static bool Rcpp::Vector< RTYPE, StoragePolicy >::is_na (
             stored_type x )  [inline], [static]
```

Definition at line 255 of file Vector.h.

6.813.4.43 length()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::length ( ) const  [inline]
```

the length of the vector, uses Rf_xlength

Definition at line 269 of file Vector.h.

6.813.4.44 offset() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
             const int & i,
             const int & j ) const  [inline]
```

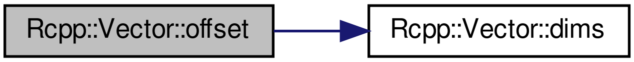
offset based on the dimensions of this vector

Definition at line 283 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::at(), Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::operator()(), Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set(), Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



6.813.4.45 offset() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
             const R_xlen_t & i ) const  [inline]
```

one dimensional offset doing bounds checking to ensure it is valid

Definition at line 303 of file Vector.h.

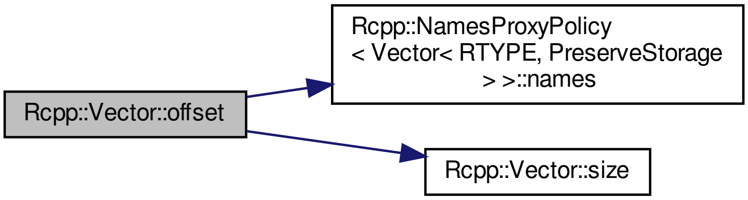
6.813.4.46 offset() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
             const std::string & name ) const  [inline]
```

Definition at line 311 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.4.47 operator RObject()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::operator RObject ( ) const  [inline]
```

Definition at line 379 of file Vector.h.

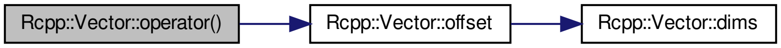
6.813.4.48 operator()() [1/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
             const size_t & i )  [inline]
```

Definition at line 344 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



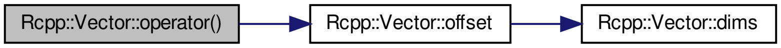
6.813.4.49 operator()() [2/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreservePolicy>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
            const size_t & i ) const  [inline]
```

Definition at line 347 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



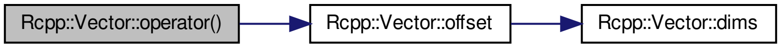
6.813.4.50 operator()() [3/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
            const size_t & i,
            const size_t & j )  [inline]
```

Definition at line 358 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



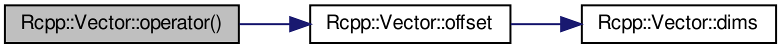
6.813.4.51 operator()() [4/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
             const size_t & i,
             const size_t & j ) const  [inline]
```

Definition at line 361 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



6.813.4.52 operator()() [5/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
             const std::string & name )  [inline]
```

Definition at line 368 of file Vector.h.

6.813.4.53 operator()() [6/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
             const std::string & name ) const  [inline]
```

Definition at line 375 of file Vector.h.

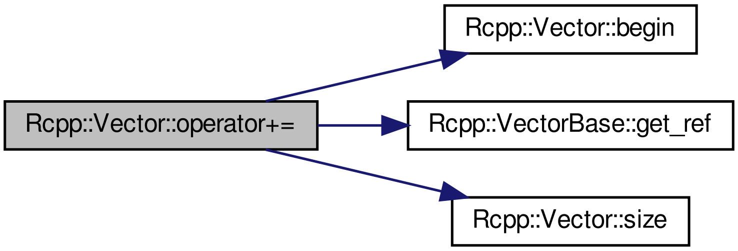
6.813.4.54 operator+=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename EXPR_VEC >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator+= (
             const VectorBase< RTYPE, false, EXPR_VEC > & rhs )  [inline]
```

Definition at line 569 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



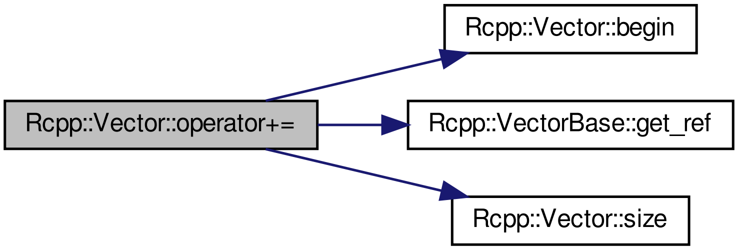
6.813.4.55 operator+=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename EXPR_VEC >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator+= (
             const VectorBase< RTYPE, true, EXPR_VEC > & rhs )  [inline]
```

Definition at line 552 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



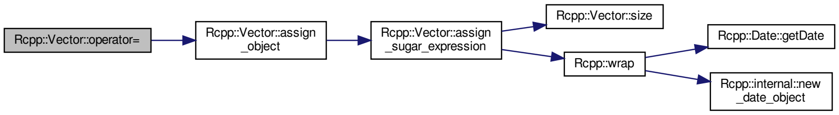
6.813.4.56 operator=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (
             const T & x )  [inline]
```

Definition at line 247 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::assign_object().

Here is the call graph for this function:


Rcpp::Vector::operator=
Rcpp::Vector::assign _object
Rcpp::Vector::assign _sugar_expression
Rcpp::Vector::size
Rcpp::wrap
Rcpp::Date::getDate
Rcpp::internal::new _date_object

6.813.4.57 operator=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (
             const Vector< RTYPE, StoragePolicy > & rhs )  [inline]
```

Definition at line 68 of file Vector.h.

Referenced by Rcpp::newDatetimeVector::operator=(), and Rcpp::newDateVector::operator=().

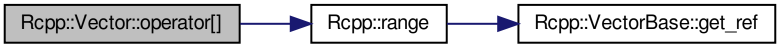
6.813.4.58 operator[]() [1/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Indexer Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
             const Range & range )  [inline]
```

Definition at line 547 of file Vector.h.

References Rcpp::range().

Here is the call graph for this function:



6.813.4.59 operator[]() [2/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
             const std::string & name )  [inline]
```

Definition at line 365 of file Vector.h.

6.813.4.60 operator[]() [3/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
             const std::string & name ) const  [inline]
```

Definition at line 372 of file Vector.h.

6.813.4.61 operator[]() [4/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
SubsetProxy<RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T> Rcpp::Vector< RTYPE, StoragePolicy
>::operator[] (
             const VectorBase< RHS_RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 386 of file Vector.h.

6.813.4.62 operator[]() [5/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
const SubsetProxy<RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T> Rcpp::Vector< RTYPE, Storage↩
Policy >::operator[] (
             const VectorBase< RHS_RTYPE, RHS_NA, RHS_T > & rhs ) const  [inline]
```

Definition at line 395 of file Vector.h.

6.813.4.63 operator[]() [6/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
             R_xlen_t i )  [inline]
```

Definition at line 341 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

6.813.4.64 operator[]() [7/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 342 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

6.813.4.65 push_back() [1/2]

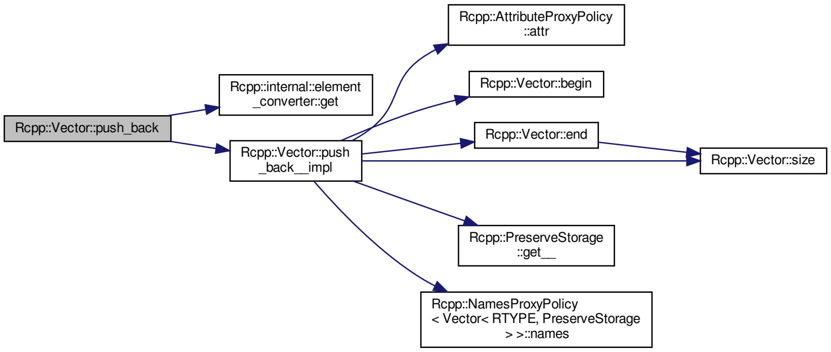
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back (
             const T & object )  [inline]
```

Definition at line 452 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl().

Referenced by Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set(), Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set(), Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:


Rcpp::Vector::push_back
Rcpp::internal::element_converter::get
Rcpp::Vector::push_back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

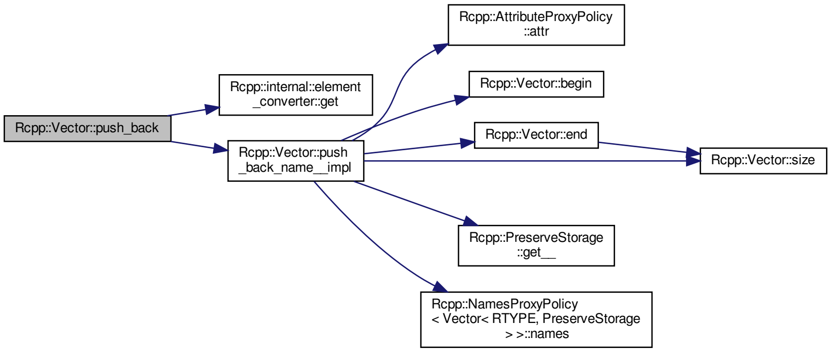
6.813.4.66 push_back() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back (
             const T & object,
             const std::string & name )  [inline]
```

Definition at line 459 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl().

Here is the call graph for this function:


Rcpp::Vector::push_back
Rcpp::internal::element_converter::get
Rcpp::Vector::push_back_name__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

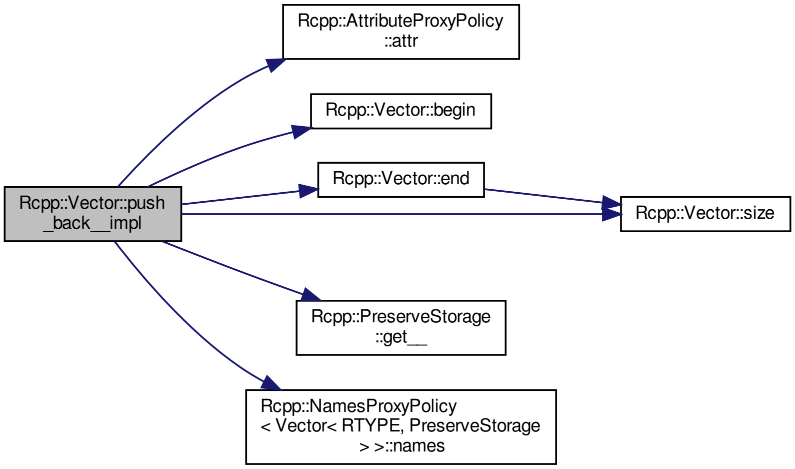
6.813.4.67 push_back__impl() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl (
            const stored_type & object,
            traits::false_type ) [inline], [private]
```

Definition at line 650 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp↩::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::push _back__impl
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

6.813.4.68 push_back__impl() [2/2]

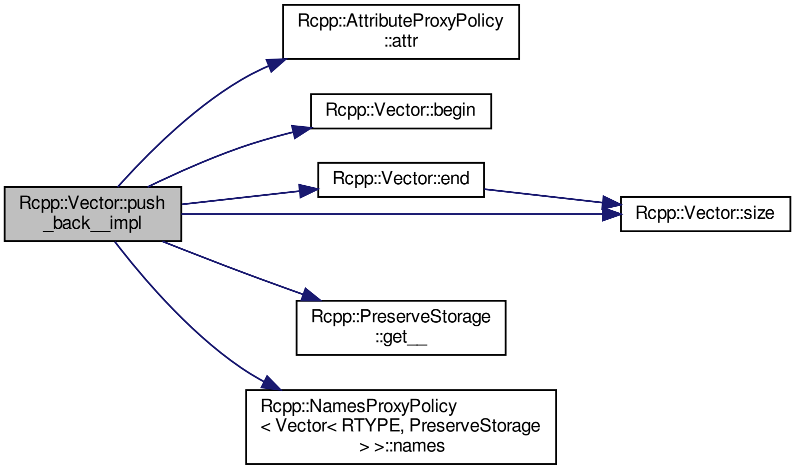
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl (
            const stored_type & object,
            traits::true_type  )  [inline], [private]
```

Definition at line 624 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Here is the call graph for this function:


Rcpp::Vector::push_back__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

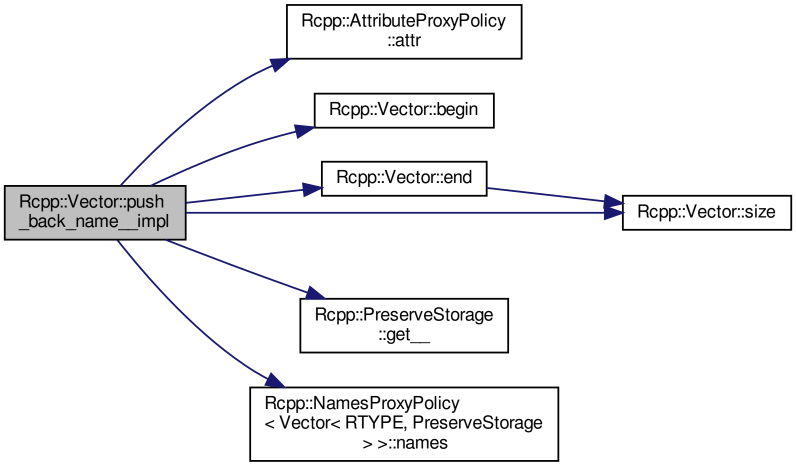
6.813.4.69 push_back_name__impl() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl (
            const stored_type & object,
            const std::string & name,
            traits::false_type ) [inline], [private]
```

Definition at line 702 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:


Rcpp::Vector::push_back_name__impl
Rcpp::AttributeProxyPolicy::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage::get__
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names

6.813.4.70 push_back_name__impl() [2/2]

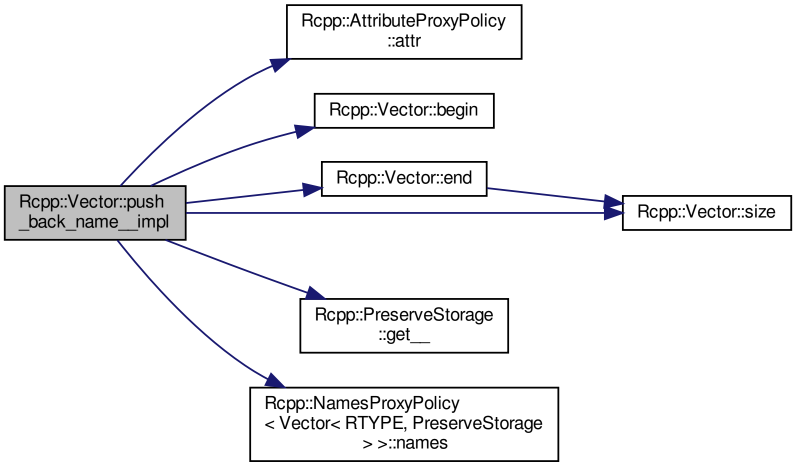
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl (
            const stored_type & object,
            const std::string & name,
            traits::true_type ) [inline], [private]
```

Definition at line 675 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push_back().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::push _back_name__impl
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

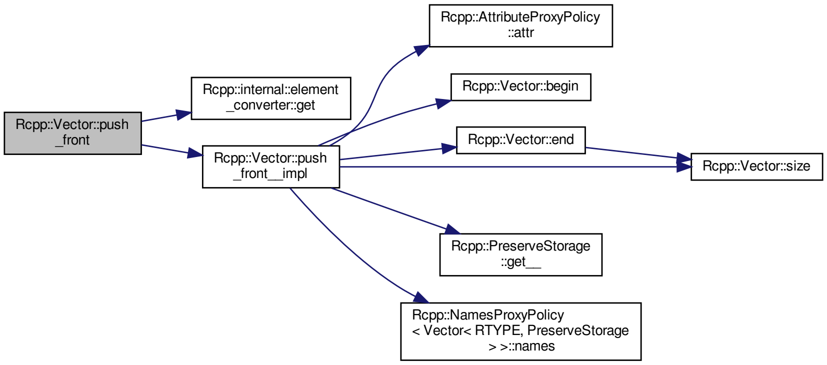
6.813.4.71 push_front() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front (
            const T & object )  [inline]
```

Definition at line 466 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl().

Here is the call graph for this function:


Rcpp::Vector::push _front
Rcpp::internal::element _converter::get
Rcpp::Vector::push _front__impl
Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::PreserveStorage ::get__
Rcpp::NamesProxyPolicy < Vector< RTYPE, PreserveStorage > >::names

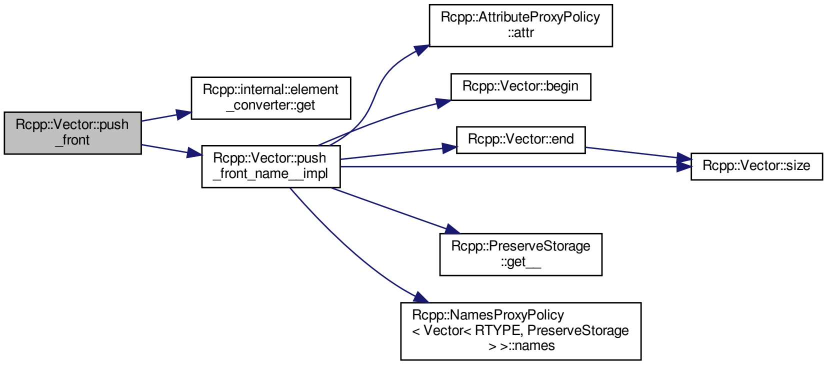
6.813.4.72 push_front() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front (
            const T & object,
            const std::string & name ) [inline]
```

Definition at line 472 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl().

Here is the call graph for this function:



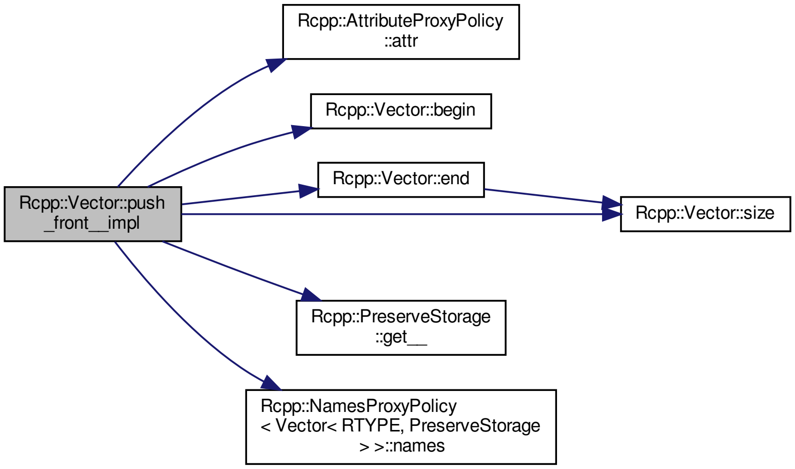
6.813.4.73 push_front__impl() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl (
            const stored_type & object,
            traits::false_type ) [inline], [private]
```

Definition at line 757 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.4.74 push_front__impl() [2/2]

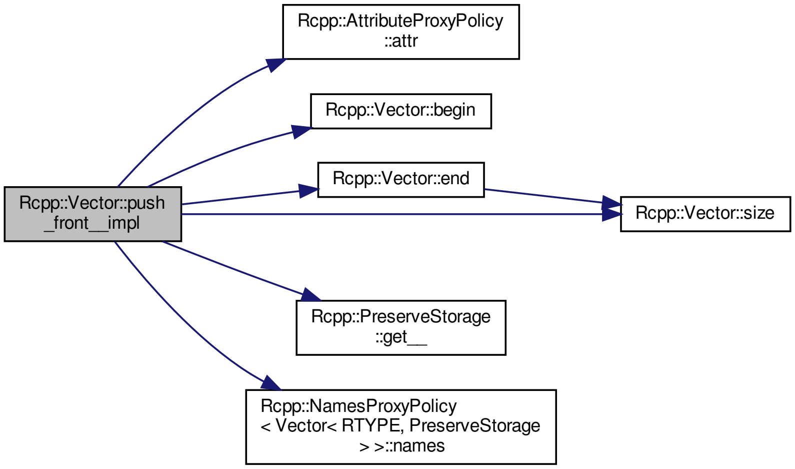
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl (
            const stored_type & object,
            traits::true_type ) [inline], [private]
```

Definition at line 730 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push_front().

Here is the call graph for this function:



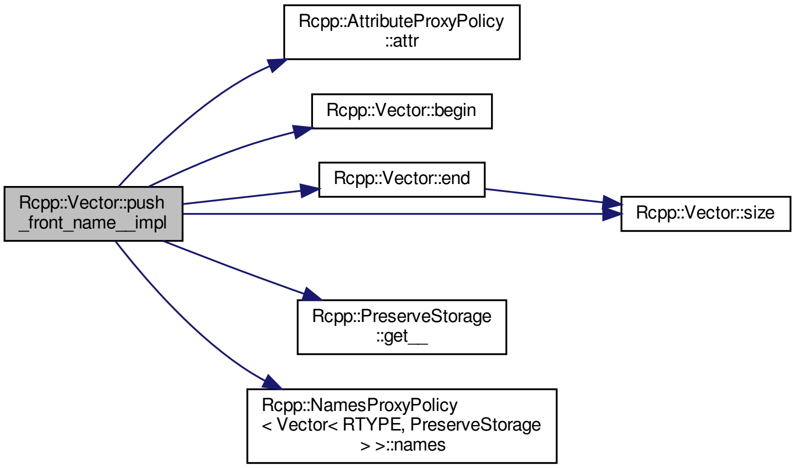
6.813.4.75 push_front_name__impl() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl (
             const stored_type & object,
             const std::string & name,
             traits::false_type  )  [inline], [private]
```

Definition at line 813 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.4.76 push_front_name__impl() [2/2]

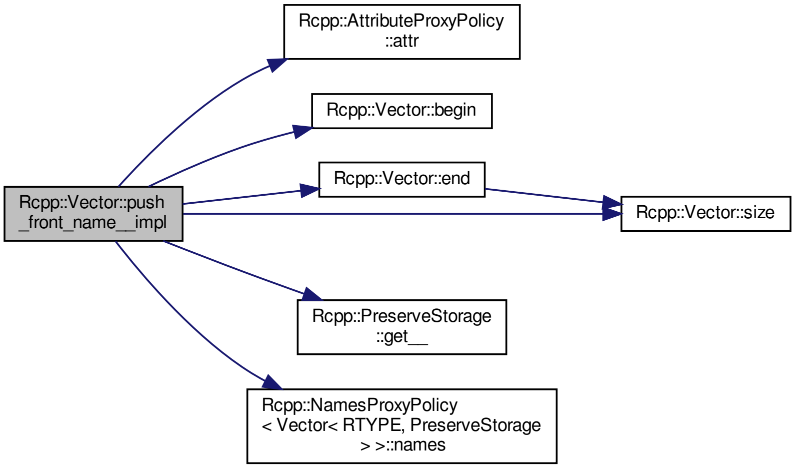
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl (
            const stored_type & object,
            const std::string & name,
            traits::true_type ) [inline], [private]
```

Definition at line 784 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get__(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP_GET_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push_front().

Here is the call graph for this function:



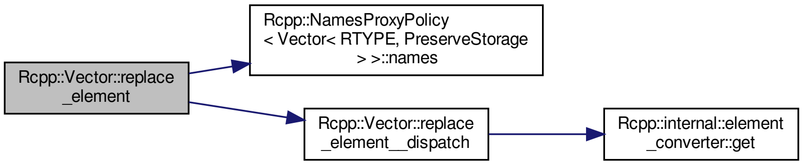
6.813.4.77 replace_element()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element (
             iterator it,
             SEXP names,
             R_xlen_t index,
             const U & u )  [inline], [static]
```

Definition at line 514 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch().

Here is the call graph for this function:



6.813.4.78 replace_element__dispatch() [1/2]

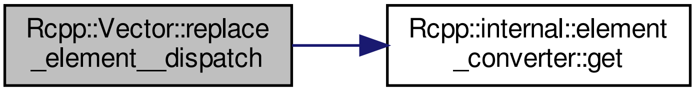
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch (
             traits::false_type ,
             iterator it,
             SEXP names,
             R_xlen_t index,
             const U & u )  [inline], [static]
```

Definition at line 520 of file Vector.h.

References Rcpp::internal::element_converter< RTYPE >::get().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::replace_element().

Here is the call graph for this function:



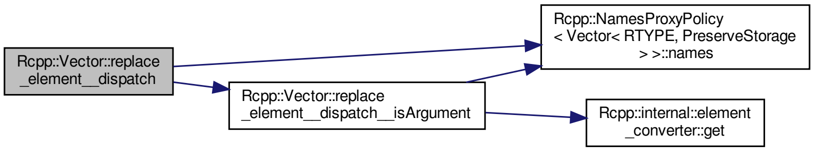
6.813.4.79 replace_element__dispatch() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch (
             traits::true_type ,
             iterator it,
             SEXP names,
             R_xlen_t index,
             const U & u )  [inline], [static]
```

Definition at line 525 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument().

Here is the call graph for this function:



6.813.4.80 replace_element__dispatch__isArgument() [1/2]

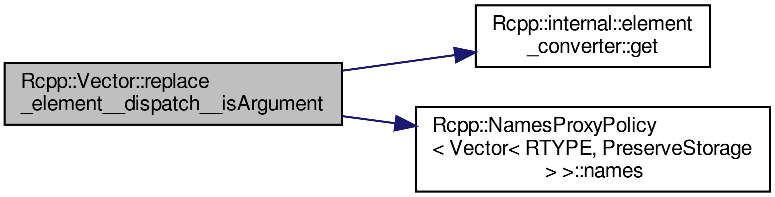
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument (
             traits::false_type ,
             iterator it,
             SEXP names,
             R_xlen_t index,
             const U & u )  [inline], [static]
```

Definition at line 530 of file Vector.h.

References DEMANGLE, Rcpp::internal::element_converter< RTYPE >::get(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and RCPP_DEBUG_2.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch().

Here is the call graph for this function:



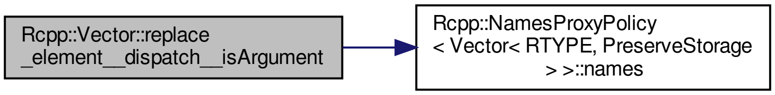
6.813.4.81 replace_element__dispatch__isArgument() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument (
             traits::true_type ,
             iterator it,
             SEXP names,
             R_xlen_t index,
             const U & u ) [inline], [static]
```

Definition at line 538 of file Vector.h.

References DEMANGLE, Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and RCPP_DEBUG_2.

Here is the call graph for this function:


Rcpp::Vector::replace
_element__dispatch__isArgument
Rcpp::NamesProxyPolicy
< Vector< RTYPE, PreserveStorage
> >::names

6.813.4.82 size()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::size ( ) const [inline]
```

alias of length

Examples

ConvolveBenchmarks/convolve3_cpp.cpp, and ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 276 of file Vector.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression(), Rcpp::Vector< RTYPE, StoragePolicy >::cend(), convolve(), convolve10cpp(), convolve11cpp(), convolve12cpp(), convolve14cpp(), convolve3cpp(), convolve4cpp(), convolve5cpp(), convolve8cpp(), convolve9cpp(), Rcpp::sugar::EmpiricalSample(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl(), fastLm(), Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch(), Rcpp::DataFrame_Impl< StoragePolicy >::from_list(), Rcpp::sugar::Cummax< RTYPE, NA, T >::get(), Rcpp::sugar::Cummin< RTYPE, NA, T >::get(), Rcpp::sugar::Cumprod< RTYPE, NA, T >::get(), Rcpp::sugar::Cumsum< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< CPLXSXP, NA, T >::get(), Rcpp::sugar::Mean< LGLSXP, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices(), Rcpp::newDateVector::getDates(), Rcpp::newDatetimeVector::getDatetimes(), Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl(), lapplyCpp(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::sugar::Normalize(), Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front__impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl(), Rcpp::sample(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::ListOf< T >::size(), Rcpp::Vector< RTYPE, StoragePolicy >::sort(), Rcpp::trimws(), Rcpp::Vector< RTYPE, StoragePolicy >::Vector(), and Rcpp::sugar::WalkerSample().

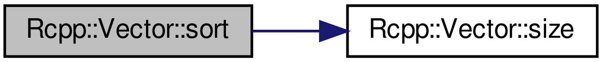
6.813.4.83 sort()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::sort (
             bool decreasing = false )  [inline]
```

Definition at line 402 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



6.813.4.84 update()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::update (
             SEXP  )  [inline]
```

Definition at line 509 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

6.813.5 Member Data Documentation

6.813.5.1 cache

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
traits::r_vector_cache_type<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy >↩
::cache
```

Definition at line 41 of file Vector.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::at(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp↩ ::Vector< RTYPE, StoragePolicy >::cbegin(), Rcpp::Vector< RTYPE, StoragePolicy >::cend(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::Vector< RTYPE, StoragePolicy >::erase(), Rcpp::Vector< RTYPE, StoragePolicy >↩ ::insert(), Rcpp::Vector< RTYPE, StoragePolicy >::operator()(), Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ](), and Rcpp::Vector< RTYPE, StoragePolicy >::update().

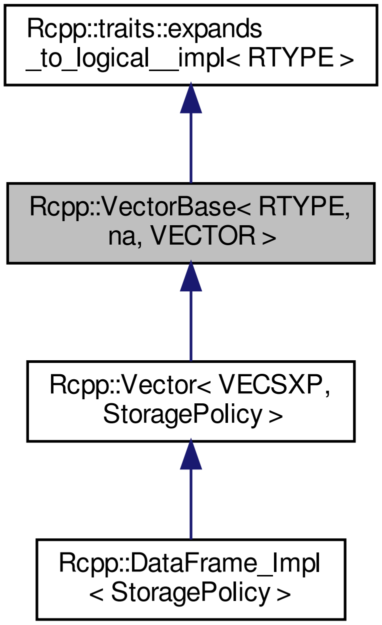
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00_forward_Vector.h
- inst/include/Rcpp/vector/Vector.h

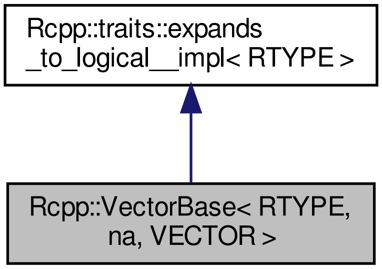
6.814 Rcpp::VectorBase< RTYPE, na, VECTOR > Class Template Reference

```
#include <VectorBase.h>
```

Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >:



Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >:



Classes

- struct can_have_na

- struct const_iter_traits
- class iter_base
- struct iter_traits
- struct r_type
- struct rcpp_sugar_expression

Public Types

- typedef traits::storage_type< RTYPE >::type stored_type
- typedef traits::storage_type< RTYPE >::type elem_type
- typedef iter_base< iter_traits > iterator
- typedef iter_base< const_iter_traits > const_iterator

Public Member Functions

- VECTOR & get_ref ()
- const VECTOR & get_ref () const
- stored_type operator[ ] (R_xlen_t i) const
- R_xlen_t size () const
- const_iterator begin () const
- const_iterator end () const
- const_iterator cbegin () const
- const_iterator cend () const

6.814.1 Detailed Description

template<int RTYPE, bool na, typename VECTOR>
class Rcpp::VectorBase< RTYPE, na, VECTOR >

a base class for vectors, modelled after the CRTP

Definition at line 29 of file VectorBase.h.

6.814.2 Member Typedef Documentation

6.814.2.1 const_iterator

```
template<int RTYPE, bool na, typename VECTOR >
typedef iter_base< const_iter_traits > Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iterator
```

Definition at line 162 of file VectorBase.h.

6.814.2.2 elem_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef traits::storage_type<RTYPE>::type Rcpp::VectorBase< RTYPE, na, VECTOR >::elem_type
```

Definition at line 35 of file VectorBase.h.

6.814.2.3 iterator

```
template<int RTYPE, bool na, typename VECTOR >
typedef iter_base< iter_traits > Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator
```

Definition at line 161 of file VectorBase.h.

6.814.2.4 stored_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef traits::storage_type<RTYPE>::type Rcpp::VectorBase< RTYPE, na, VECTOR >::stored_type
```

Definition at line 34 of file VectorBase.h.

6.814.3 Member Function Documentation

6.814.3.1 begin()

```
template<int RTYPE, bool na, typename VECTOR >
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::begin ( ) const  [inline]
```

Definition at line 164 of file VectorBase.h.

6.814.3.2 cbegin()

```
template<int RTYPE, bool na, typename VECTOR >
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::cbegin ( ) const  [inline]
```

Definition at line 167 of file VectorBase.h.

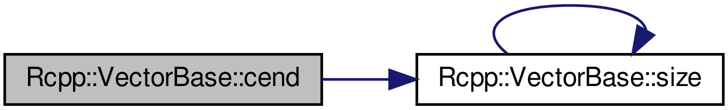
6.814.3.3 cend()

```
template<int RTYPE, bool na, typename VECTOR >
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::cend ( ) const  [inline]
```

Definition at line 168 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



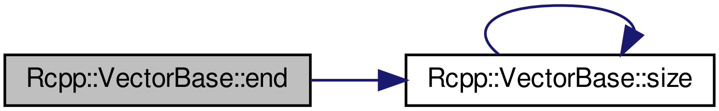
6.814.3.4 end()

```
template<int RTYPE, bool na, typename VECTOR >
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::end ( ) const  [inline]
```

Definition at line 165 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.814.3.5 get_ref() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref ( )  [inline]
```

Definition at line 37 of file VectorBase.h.

Referenced by Rcpp::clamp(), Rcpp::collapse(), Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression(), Rcpp::in(), Rcpp::intersect(), Rcpp::is_finite(), Rcpp::is_infinite(), Rcpp::is_nan(), Rcpp::mapply(), Rcpp::match(), Rcpp::max(), Rcpp::min(), Rcpp::na_omit(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::MatrixColumn< RTYPE >::operator=(), Rcpp::MatrixRow< RTYPE >::operator=(), Rcpp::pmax(), Rcpp::pmin(), Rcpp::pow(), Rcpp::range(), Rcpp::setdiff(), Rcpp::setequal(), Rcpp::table(), and Rcpp::union_().

6.814.3.6 get_ref() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
const VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref ( ) const  [inline]
```

Definition at line 41 of file VectorBase.h.

6.814.3.7 operator[]()

```
template<int RTYPE, bool na, typename VECTOR >
stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 45 of file VectorBase.h.

6.814.3.8 size()

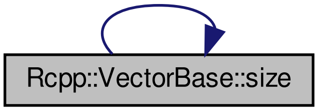
```
template<int RTYPE, bool na, typename VECTOR >
R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::size ( ) const  [inline]
```

Definition at line 49 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::VectorBase< RTYPE, na, VECTOR >::cend(), Rcpp::collapse(), Rcpp::VectorBase< RTYPE, na, VECTOR >::end(), Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMax< RTYPE, false, T >::get(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMin< RTYPE, false, T >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression(), Rcpp::seq_along(), Rcpp::stats::D0< RTYPE, NA, T >::size(), Rcpp::stats::D1< RTYPE, NA, T >::size(), Rcpp::stats::D2< RTYPE, NA, T >::size(), Rcpp::stats::D3< RTYPE, NA, T >::size(), Rcpp::stats::P0< RTYPE, NA, T >::size(), Rcpp::stats::P1< RTYPE, NA, T >::size(), Rcpp::stats::P2< RTYPE, NA, T >::size(), Rcpp::stats::P3< RTYPE, NA, T >::size(), Rcpp::stats::Q0< RTYPE, NA, T >::size(), Rcpp::stats::Q1< RTYPE, NA, T >::size(), Rcpp::stats::Q2< RTYPE, NA, T >::size(), Rcpp::stats::Q3< RTYPE, NA, T >::size(), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::size(), Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size(), Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size(), Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size(), Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size(), Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::size(), Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::size(), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::size(), Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::size(), Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size(), Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size(), Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size(), Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::size(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::size(), Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size(), Rcpp::sugar::Not_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::size(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



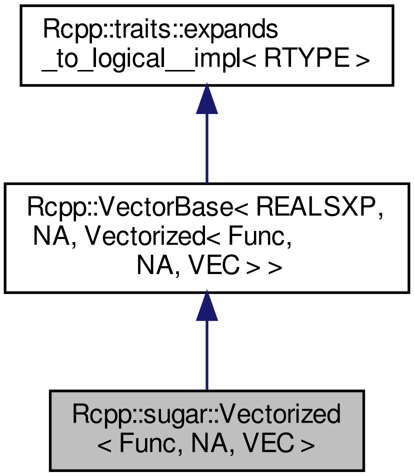
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/VectorBase.h

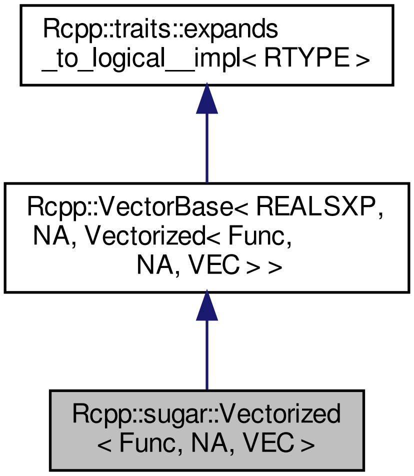
6.815 Rcpp::sugar::Vectorized< Func, NA, VEC > Class Template Reference

```
#include <Vectorized_Math.h>
```

Inheritance diagram for Rcpp::sugar::Vectorized< Func, NA, VEC >:



Collaboration diagram for Rcpp::sugar::Vectorized< Func, NA, VEC >:



Public Types

- typedef Rcpp::VectorBase< REALSXP, NA, VEC > VEC_TYPE

- typedef Rcpp::traits::Extractor< REALSXP, NA, VEC >::type VEC_EXT

Public Member Functions

- Vectorized (const VEC_TYPE &object_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & object

6.815.1 Detailed Description

template<DDFun Func, bool NA, typename VEC>
class Rcpp::sugar::Vectorized< Func, NA, VEC >

Definition at line 31 of file Vectorized_Math.h.

6.815.2 Member Typedef Documentation

6.815.2.1 VEC_EXT

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::traits::Extractor<REALSXP,NA,VEC>::type Rcpp::sugar::Vectorized< Func, NA, VEC >↩
::VEC_EXT
```

Definition at line 34 of file Vectorized_Math.h.

6.815.2.2 VEC_TYPE

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::VectorBase<REALSXP,NA,VEC> Rcpp::sugar::Vectorized< Func, NA, VEC >::VEC_TYPE
```

Definition at line 33 of file Vectorized_Math.h.

6.815.3 Constructor & Destructor Documentation

6.815.3.1 Vectorized()

```
template<DDFun Func, bool NA, typename VEC >
Rcpp::sugar::Vectorized< Func, NA, VEC >::Vectorized (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 36 of file Vectorized_Math.h.

6.815.4 Member Function Documentation

6.815.4.1 operator[]()

```
template<DDFun Func, bool NA, typename VEC >
double Rcpp::sugar::Vectorized< Func, NA, VEC >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 37 of file Vectorized_Math.h.

6.815.4.2 size()

```
template<DDFun Func, bool NA, typename VEC >
R_xlen_t Rcpp::sugar::Vectorized< Func, NA, VEC >::size ( ) const  [inline]
```

Definition at line 40 of file Vectorized_Math.h.

6.815.5 Member Data Documentation

6.815.5.1 object

```
template<DDFun Func, bool NA, typename VEC >
const VEC_EXT& Rcpp::sugar::Vectorized< Func, NA, VEC >::object  [private]
```

Definition at line 43 of file Vectorized_Math.h.

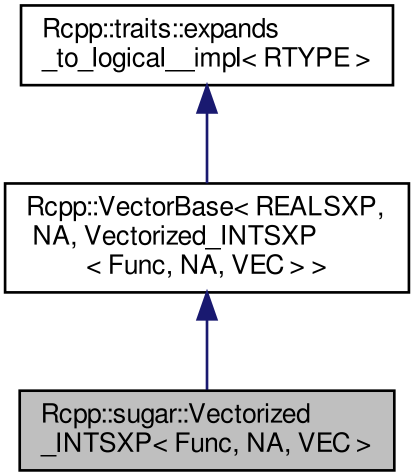
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized_Math.h

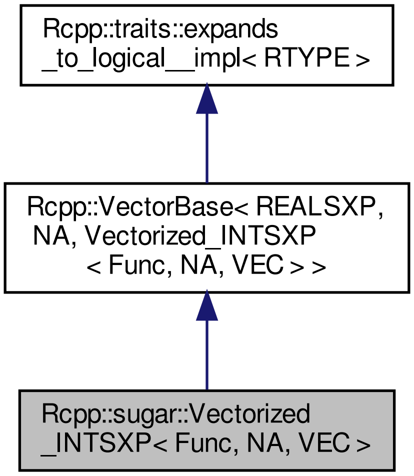
6.816 Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC > Class Template Reference

```
#include <Vectorized_Math.h>
```

Inheritance diagram for Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >:



Collaboration diagram for Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >:


Rcpp::traits::expands_to_logical__impl< RTYPE >
Rcpp::VectorBase< REALSXP, NA, Vectorized_INTSXP< Func, NA, VEC > >
Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >

Public Types

- typedef Rcpp::VectorBase< INTSXP, NA, VEC > VEC_TYPE
- typedef Rcpp::traits::Extractor< INTSXP, NA, VEC >::type VEC_EXT

Public Member Functions

- Vectorized_INTSXP (const VEC_TYPE &object_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & object

6.816.1 Detailed Description

template<DDFun Func, bool NA, typename VEC>
class Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >

Definition at line 47 of file Vectorized_Math.h.

6.816.2 Member Typedef Documentation

6.816.2.1 VEC_EXT

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::traits::Extractor<INTSXP,NA,VEC>::type Rcpp::sugar::Vectorized_INTSXP< Func, NA,
VEC >::VEC_EXT
```

Definition at line 50 of file Vectorized_Math.h.

6.816.2.2 VEC_TYPE

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::VectorBase<INTSXP,NA,VEC> Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::VEC_TYPE
```

Definition at line 49 of file Vectorized_Math.h.

6.816.3 Constructor & Destructor Documentation

6.816.3.1 Vectorized_INTSXP()

```
template<DDFun Func, bool NA, typename VEC >
Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::Vectorized_INTSXP (
             const VEC_TYPE & object_ )  [inline]
```

Definition at line 52 of file Vectorized_Math.h.

6.816.4 Member Function Documentation

6.816.4.1 operator[]()

```
template<DDFun Func, bool NA, typename VEC >
double Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::operator[] (
             R_xlen_t i ) const  [inline]
```

Definition at line 53 of file Vectorized_Math.h.

6.816.4.2 size()

```
template<DDFun Func, bool NA, typename VEC >
R_xlen_t Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::size ( ) const  [inline]
```

Definition at line 58 of file Vectorized_Math.h.

6.816.5 Member Data Documentation

6.816.5.1 object

```
template<DDFun Func, bool NA, typename VEC >
const VEC_EXT& Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::object  [private]
```

Definition at line 61 of file Vectorized_Math.h.

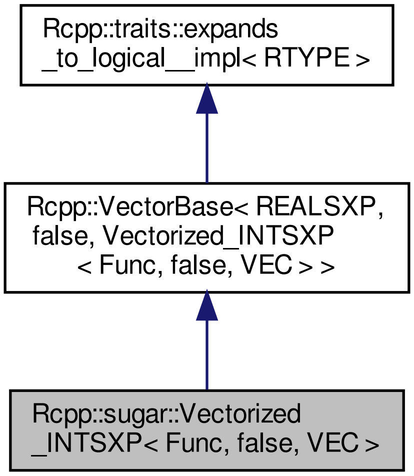
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized_Math.h

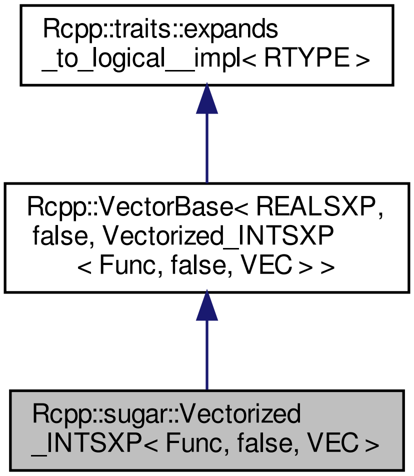
6.817 Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC > Class Template Reference

```
#include <Vectorized_Math.h>
```

Inheritance diagram for Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >:



Collaboration diagram for Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >:



Public Types

- typedef Rcpp::VectorBase< INTSXP, false, VEC > VEC_TYPE
- typedef Rcpp::traits::Extractor< INTSXP, false, VEC >::type VEC_EXT

Public Member Functions

- Vectorized_INTSXP (const VEC_TYPE &object_)
- double operator[ ] (R_xlen_t i) const
- R_xlen_t size () const

Private Attributes

- const VEC_EXT & object

6.817.1 Detailed Description

template<DDFun Func, typename VEC>
class Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >

Definition at line 64 of file Vectorized_Math.h.

6.817.2 Member Typedef Documentation

6.817.2.1 VEC_EXT

```
template<DDFun Func, typename VEC >
typedef Rcpp::traits::Extractor<INTSXP,false,VEC>::type Rcpp::sugar::Vectorized_INTSXP< Func,
false, VEC >::VEC_EXT
```

Definition at line 68 of file Vectorized_Math.h.

6.817.2.2 VEC_TYPE

```
template<DDFun Func, typename VEC >
typedef Rcpp::VectorBase<INTSXP,false,VEC> Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >↩
::VEC_TYPE
```

Definition at line 67 of file Vectorized_Math.h.

6.817.3 Constructor & Destructor Documentation

6.817.3.1 Vectorized_INTSXP()

```
template<DDFun Func, typename VEC >
Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::Vectorized_INTSXP (
                const VEC_TYPE & object_ )  [inline]
```

Definition at line 70 of file Vectorized_Math.h.

6.817.4 Member Function Documentation

6.817.4.1 operator[]()

```
template<DDFun Func, typename VEC >
double Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::operator[] (
                R_xlen_t i ) const  [inline]
```

Definition at line 71 of file Vectorized_Math.h.

6.817.4.2 size()

```
template<DDFun Func, typename VEC >
R_xlen_t Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::size ( ) const  [inline]
```

Definition at line 74 of file Vectorized_Math.h.

6.817.5 Member Data Documentation

6.817.5.1 object

```
template<DDFun Func, typename VEC >
const VEC_EXT& Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::object  [private]
```

Definition at line 77 of file Vectorized_Math.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized_Math.h

6.818 Rcpp::void_type Struct Reference

```
#include <get_return_type.h>
```

6.818.1 Detailed Description

Definition at line 27 of file get_return_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/module/get_return_type.h

6.819 Rcpp::traits::void_wrap_tag Struct Reference

```
#include <module_wrap_traits.h>
```

6.819.1 Detailed Description

Definition at line 30 of file module_wrap_traits.h.

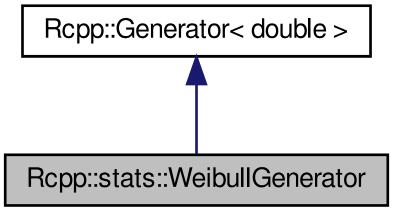
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/module_wrap_traits.h

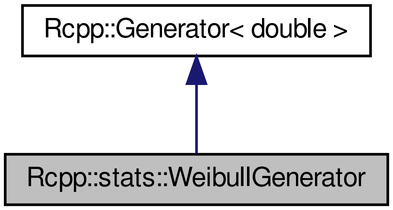
6.820 Rcpp::stats::WeibullGenerator Class Reference

```
#include <rweibull.h>
```

Inheritance diagram for Rcpp::stats::WeibullGenerator:



Collaboration diagram for Rcpp::stats::WeibullGenerator:



Public Member Functions

- WeibullGenerator (double shape_, double scale_)
- double operator() () const

Private Attributes

- double shape_inv
- double scale

Additional Inherited Members

6.820.1 Detailed Description

Definition at line 28 of file rweibull.h.

6.820.2 Constructor & Destructor Documentation

6.820.2.1 WeibullGenerator()

```
Rcpp::stats::WeibullGenerator::WeibullGenerator (
             double shape_,
             double scale_ )  [inline]
```

Definition at line 31 of file rweibull.h.

6.820.3 Member Function Documentation

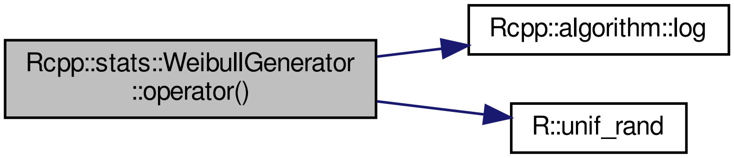
6.820.3.1 operator()()

```
double Rcpp::stats::WeibullGenerator::operator() ( ) const  [inline]
```

Definition at line 34 of file rweibull.h.

References Rcpp::algorithm::log(), scale, shape_inv, and R::unif_rand().

Here is the call graph for this function:



6.820.4 Member Data Documentation

6.820.4.1 scale

```
double Rcpp::stats::WeibullGenerator::scale  [private]
```

Definition at line 39 of file rweibull.h.

Referenced by operator()().

6.820.4.2 shape_inv

```
double Rcpp::stats::WeibullGenerator::shape_inv  [private]
```

Definition at line 39 of file rweibull.h.

Referenced by operator()().

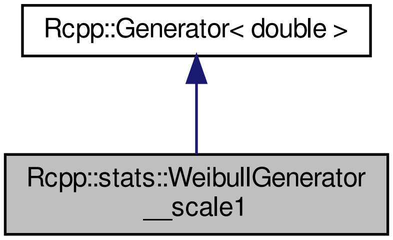
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rweibull.h

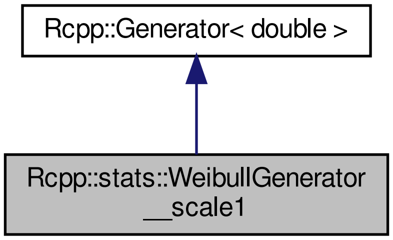
6.821 Rcpp::stats::WeibullGenerator__scale1 Class Reference

```
#include <rweibull.h>
```

Inheritance diagram for Rcpp::stats::WeibullGenerator__scale1:


Rcpp::Generator< double >
Rcpp::stats::WeibullGenerator __scale1

Collaboration diagram for Rcpp::stats::WeibullGenerator__scale1:



Public Member Functions

- WeibullGenerator__scale1 (double shape_)
- double operator() () const

Private Attributes

- double shape_inv

Additional Inherited Members

6.821.1 Detailed Description

Definition at line 42 of file rweibull.h.

6.821.2 Constructor & Destructor Documentation

6.821.2.1 WeibullGenerator__scale1()

```
Rcpp::stats::WeibullGenerator__scale1::WeibullGenerator__scale1 (
            double shape_ )  [inline]
```

Definition at line 45 of file rweibull.h.

6.821.3 Member Function Documentation

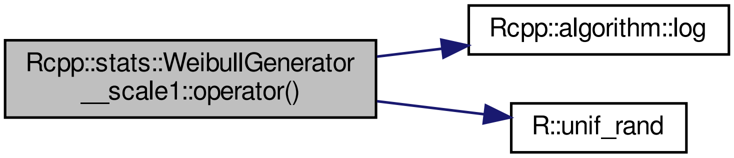
6.821.3.1 operator()()

```
double Rcpp::stats::WeibullGenerator__scale1::operator() ( ) const  [inline]
```

Definition at line 48 of file rweibull.h.

References Rcpp::algorithm::log(), shape_inv, and R::unif_rand().

Here is the call graph for this function:



6.821.4 Member Data Documentation

6.821.4.1 shape_inv

```
double Rcpp::stats::WeibullGenerator__scale1::shape_inv  [private]
```

Definition at line 53 of file rweibull.h.

Referenced by operator()().

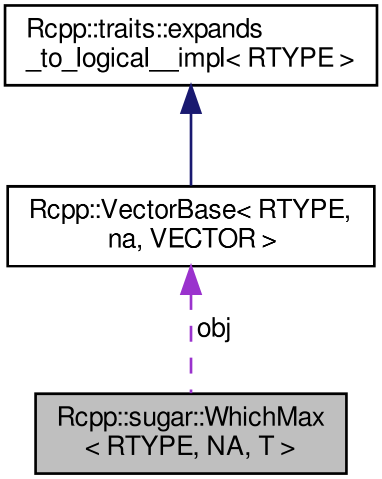
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rweibull.h

6.822 Rcpp::sugar::WhichMax< RTYPE, NA, T > Class Template Reference

```
#include <which_max.h>
```

Collaboration diagram for Rcpp::sugar::WhichMax< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- WhichMax (const VEC_TYPE &obj_)
- R_xlen_t get () const

Private Attributes

- const VEC_TYPE & obj

6.822.1 Detailed Description

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::WhichMax< RTYPE, NA, T >

Definition at line 29 of file which_max.h.

6.822.2 Member Typedef Documentation

6.822.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMax< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file which_max.h.

6.822.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::WhichMax< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file which_max.h.

6.822.3 Constructor & Destructor Documentation

6.822.3.1 WhichMax()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::WhichMax< RTYPE, NA, T >::WhichMax (
             const VEC_TYPE & obj_ )  [inline]
```

Definition at line 33 of file which_max.h.

6.822.4 Member Function Documentation

6.822.4.1 get()

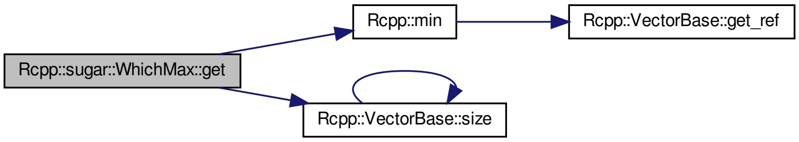
```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::WhichMax< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 35 of file which_max.h.

References Rcpp::min(), Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::which_max().

Here is the call graph for this function:



6.822.5 Member Data Documentation

6.822.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj [private]
```

Definition at line 53 of file which_max.h.

Referenced by Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), and Rcpp::sugar::WhichMax< RTYPE, false, T >↩
::get().

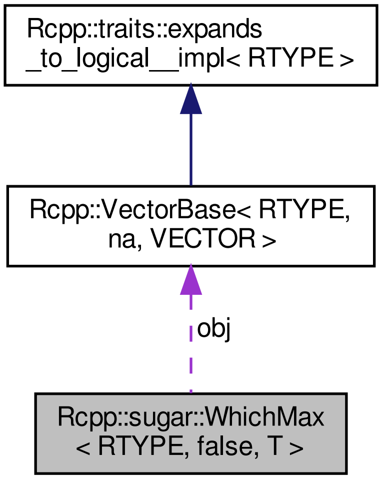
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_max.h

6.823 Rcpp::sugar::WhichMax< RTYPE, false, T > Class Template Reference

```
#include <which_max.h>
```

Collaboration diagram for Rcpp::sugar::WhichMax< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- WhichMax (const VEC_TYPE &obj_)
- R_xlen_t get () const

Private Attributes

- const VEC_TYPE & obj

6.823.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::WhichMax< RTYPE, false, T >

Definition at line 58 of file which_max.h.

6.823.2 Member Typedef Documentation

6.823.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMax< RTYPE, false, T >::STORAGE
```

Definition at line 61 of file which_max.h.

6.823.2.2 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::WhichMax< RTYPE, false, T >::VEC_TYPE
```

Definition at line 60 of file which_max.h.

6.823.3 Constructor & Destructor Documentation

6.823.3.1 WhichMax()

```
template<int RTYPE, typename T >
Rcpp::sugar::WhichMax< RTYPE, false, T >::WhichMax (
                const VEC_TYPE & obj_ )  [inline]
```

Definition at line 62 of file which_max.h.

6.823.4 Member Function Documentation

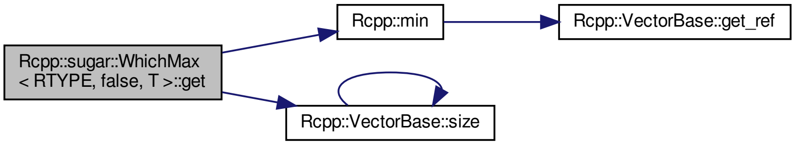
6.823.4.1 get()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::WhichMax< RTYPE, false, T >::get ( ) const  [inline]
```

Definition at line 64 of file which_max.h.

References Rcpp::min(), Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.823.5 Member Data Documentation

6.823.5.1 obj

```
template<int RTYPE, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMax< RTYPE, false, T >::obj  [private]
```

Definition at line 80 of file which_max.h.

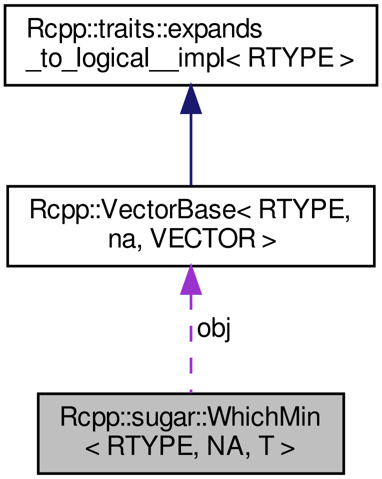
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_max.h

6.824 Rcpp::sugar::WhichMin< RTYPE, NA, T > Class Template Reference

```
#include <which_min.h>
```

Collaboration diagram for Rcpp::sugar::WhichMin< RTYPE, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- WhichMin (const VEC_TYPE &obj_)
- R_xlen_t get () const

Private Attributes

- const VEC_TYPE & obj

6.824.1 Detailed Description

**template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::WhichMin< RTYPE, NA, T >**

Definition at line 29 of file which_min.h.

6.824.2 Member Typedef Documentation

6.824.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMin< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file which_min.h.

6.824.2.2 VEC_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::WhichMin< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file which_min.h.

6.824.3 Constructor & Destructor Documentation

6.824.3.1 WhichMin()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::WhichMin< RTYPE, NA, T >::WhichMin (
              const VEC_TYPE & obj_ )  [inline]
```

Definition at line 33 of file which_min.h.

6.824.4 Member Function Documentation

6.824.4.1 get()

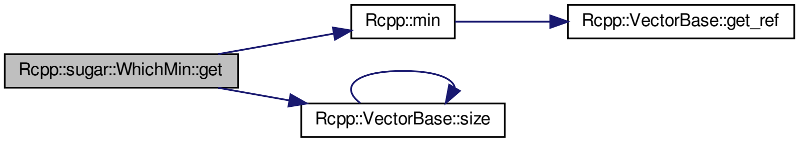
```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::WhichMin< RTYPE, NA, T >::get ( ) const  [inline]
```

Definition at line 35 of file which_min.h.

References Rcpp::min(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::which_min().

Here is the call graph for this function:



6.824.5 Member Data Documentation

6.824.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj  [private]
```

Definition at line 53 of file which_min.h.

Referenced by Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), and Rcpp::sugar::WhichMin< RTYPE, false, T >↩ ::get().

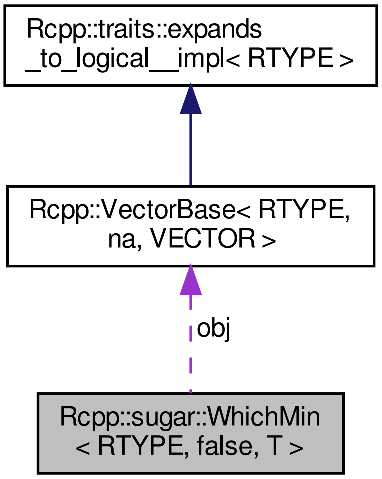
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_min.h

6.825 Rcpp::sugar::WhichMin< RTYPE, false, T > Class Template Reference

```
#include <which_min.h>
```

Collaboration diagram for Rcpp::sugar::WhichMin< RTYPE, false, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, T > VEC_TYPE
- typedef Rcpp::traits::storage_type< RTYPE >::type STORAGE

Public Member Functions

- WhichMin (const VEC_TYPE &obj_)
- R_xlen_t get () const

Private Attributes

- const VEC_TYPE & obj

6.825.1 Detailed Description

template<int RTYPE, typename T>
class Rcpp::sugar::WhichMin< RTYPE, false, T >

Definition at line 58 of file which_min.h.

6.825.2 Member Typedef Documentation

6.825.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMin< RTYPE, false, T >::STORAGE
```

Definition at line 61 of file which_min.h.

6.825.2.2 VEC_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::WhichMin< RTYPE, false, T >::VEC_TYPE
```

Definition at line 60 of file which_min.h.

6.825.3 Constructor & Destructor Documentation

6.825.3.1 WhichMin()

```
template<int RTYPE, typename T >
Rcpp::sugar::WhichMin< RTYPE, false, T >::WhichMin (
                const VEC_TYPE & obj_ )  [inline]
```

Definition at line 62 of file which_min.h.

6.825.4 Member Function Documentation

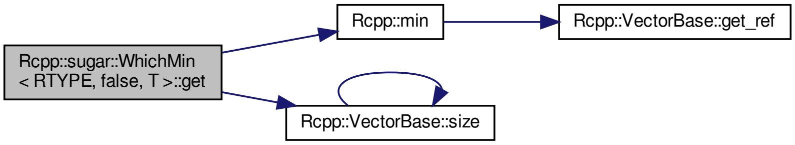
6.825.4.1 get()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::WhichMin< RTYPE, false, T >::get ( ) const  [inline]
```

Definition at line 64 of file which_min.h.

References Rcpp::min(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



6.825.5 Member Data Documentation

6.825.5.1 obj

```
template<int RTYPE, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMin< RTYPE, false, T >::obj  [private]
```

Definition at line 80 of file which_min.h.

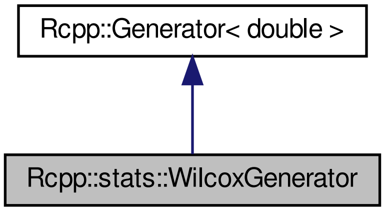
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which_min.h

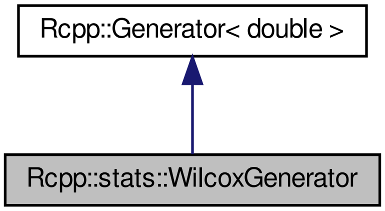
6.826 Rcpp::stats::WilcoxGenerator Class Reference

```
#include <rwilcox.h>
```

Inheritance diagram for Rcpp::stats::WilcoxGenerator:



Collaboration diagram for Rcpp::stats::WilcoxGenerator:



Public Member Functions

- WilcoxGenerator (double mm_, double nn_)
- double operator() () const

Private Attributes

- double mm
- double nn

Additional Inherited Members

6.826.1 Detailed Description

Definition at line 28 of file rwilcox.h.

6.826.2 Constructor & Destructor Documentation

6.826.2.1 WilcoxGenerator()

```
Rcpp::stats::WilcoxGenerator::WilcoxGenerator (
             double mm_,
             double nn_ )  [inline]
```

Definition at line 30 of file rwilcox.h.

6.826.3 Member Function Documentation

6.826.3.1 operator()()

```
double Rcpp::stats::WilcoxGenerator::operator() ( ) const  [inline]
```

Definition at line 31 of file rwilcox.h.

References mm, and nn.

6.826.4 Member Data Documentation

6.826.4.1 mm

```
double Rcpp::stats::WilcoxGenerator::mm  [private]
```

Definition at line 33 of file rwilcox.h.

Referenced by operator()().

6.826.4.2 nn

```
double Rcpp::stats::WilcoxGenerator::nn  [private]
```

Definition at line 33 of file rwilcox.h.

Referenced by operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rwilcox.h

6.827 Rcpp::traits::wrap_type_char_array Struct Reference

```
#include <wrap_type_traits.h>
```

6.827.1 Detailed Description

character arrays

Definition at line 57 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.828 Rcpp::traits::wrap_type_enum_tag Struct Reference

```
#include <wrap_type_traits.h>
```

6.828.1 Detailed Description

enums

Definition at line 52 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.829 Rcpp::traits::wrap_type_module_object_pointer_tag Struct Reference

```
#include <wrap_type_traits.h>
```

6.829.1 Detailed Description

module objects pointers (object<T>). conversion done using make_new_object<>()

Definition at line 42 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.830 Rcpp::traits::wrap_type_module_object_tag Struct Reference

```
#include <wrap_type_traits.h>
```

6.830.1 Detailed Description

declared module object type (by the RCPP_EXPOSED_CLASS macro)

Definition at line 47 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.831 Rcpp::traits::wrap_type_primitive_tag Struct Reference

```
#include <wrap_type_traits.h>
```

6.831.1 Detailed Description

primitive type : int, double, std::string, Rcomplex, size_t, Rbyte

Definition at line 32 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.832 Rcpp::traits::wrap_type_traits< T > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_unknown_tag wrap_category

6.832.1 Detailed Description

template<typename T>
struct Rcpp::traits::wrap_type_traits< T >

Type trait that helps the dispatch of wrap to the proper method

This builds a struct that contains a typedef called wrap_category that has to be one of "wrap_type_primitive_tag" or "wrap_type_unknown_tag"

The default is "wrap_type_unknown_tag" and this is specialized for primitive types

Definition at line 68 of file wrap_type_traits.h.

6.832.2 Member Typedef Documentation

6.832.2.1 wrap_category

```
template<typename T >
typedef wrap_type_unknown_tag Rcpp::traits::wrap_type_traits< T >::wrap_category
```

Definition at line 68 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.833 Rcpp::traits::wrap_type_traits< bool > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.833.1 Detailed Description

Definition at line 82 of file wrap_type_traits.h.

6.833.2 Member Typedef Documentation

6.833.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< bool >::wrap_category
```

Definition at line 82 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.834 Rcpp::traits::wrap_type_traits< char > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.834.1 Detailed Description

Definition at line 86 of file wrap_type_traits.h.

6.834.2 Member Typedef Documentation

6.834.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< char >::wrap_category
```

Definition at line 86 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.835 Rcpp::traits::wrap_type_traits< char[N]> Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_char_array wrap_category

6.835.1 Detailed Description

template<std::size_t N>
struct Rcpp::traits::wrap_type_traits< char[N]>

Definition at line 70 of file wrap_type_traits.h.

6.835.2 Member Typedef Documentation

6.835.2.1 wrap_category

```
template<std::size_t N>
typedef wrap_type_char_array Rcpp::traits::wrap_type_traits< char[N]>::wrap_category
```

Definition at line 70 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.836 Rcpp::traits::wrap_type_traits< const char[N]> Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_char_array wrap_category

6.836.1 Detailed Description

template<std::size_t N>
struct Rcpp::traits::wrap_type_traits< const char[N]>

Definition at line 71 of file wrap_type_traits.h.

6.836.2 Member Typedef Documentation

6.836.2.1 wrap_category

```
template<std::size_t N>
typedef wrap_type_char_array Rcpp::traits::wrap_type_traits< const char[N]>::wrap_category
```

Definition at line 71 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.837 Rcpp::traits::wrap_type_traits< const int > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.837.1 Detailed Description

Definition at line 76 of file wrap_type_traits.h.

6.837.2 Member Typedef Documentation

6.837.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< const int >::wrap_category
```

Definition at line 76 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.838 Rcpp::traits::wrap_type_traits< double > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.838.1 Detailed Description

Definition at line 78 of file wrap_type_traits.h.

6.838.2 Member Typedef Documentation

6.838.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< double >::wrap_category
```

Definition at line 78 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.839 Rcpp::traits::wrap_type_traits< float > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.839.1 Detailed Description

Definition at line 89 of file wrap_type_traits.h.

6.839.2 Member Typedef Documentation

6.839.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< float >::wrap_category
```

Definition at line 89 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.840 Rcpp::traits::wrap_type_traits< int > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.840.1 Detailed Description

Total specialization for primitive types

Definition at line 75 of file wrap_type_traits.h.

6.840.2 Member Typedef Documentation

6.840.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< int >::wrap_category
```

Definition at line 75 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.841 Rcpp::traits::wrap_type_traits< long > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.841.1 Detailed Description

Definition at line 93 of file wrap_type_traits.h.

6.841.2 Member Typedef Documentation

6.841.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< long >::wrap_category
```

Definition at line 93 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.842 Rcpp::traits::wrap_type_traits< long double > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.842.1 Detailed Description

Definition at line 96 of file wrap_type_traits.h.

6.842.2 Member Typedef Documentation

6.842.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< long double >::wrap_category
```

Definition at line 96 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.843 Rcpp::traits::wrap_type_traits< Rbyte > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.843.1 Detailed Description

Definition at line 79 of file wrap_type_traits.h.

6.843.2 Member Typedef Documentation

6.843.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rbyte >::wrap_category
```

Definition at line 79 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.844 Rcpp::traits::wrap_type_traits< Rcomplex > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.844.1 Detailed Description

Definition at line 80 of file wrap_type_traits.h.

6.844.2 Member Typedef Documentation

6.844.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcomplex >::wrap_category
```

Definition at line 80 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.845 Rcpp::traits::wrap_type_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.845.1 Detailed Description

Definition at line 28 of file Date.h.

6.845.2 Member Typedef Documentation

6.845.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcpp::Date >::wrap_category
```

Definition at line 29 of file Date.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Date.h

6.846 Rcpp::traits::wrap_type_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.846.1 Detailed Description

Definition at line 28 of file Datetime.h.

6.846.2 Member Typedef Documentation

6.846.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcpp::Datetime >::wrap_category
```

Definition at line 29 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

6.847 Rcpp::traits::wrap_type_traits< Rcpp::object< T > > Struct Template Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_module_object_pointer_tag wrap_category

6.847.1 Detailed Description

template<typename T>
struct Rcpp::traits::wrap_type_traits< Rcpp::object< T > >

Definition at line 101 of file wrap_type_traits.h.

6.847.2 Member Typedef Documentation

6.847.2.1 wrap_category

```
template<typename T >
typedef wrap_type_module_object_pointer_tag Rcpp::traits::wrap_type_traits< Rcpp::object< T >
>::wrap_category
```

Definition at line 101 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.848 Rcpp::traits::wrap_type_traits< Rcpp::String > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.848.1 Detailed Description

Definition at line 85 of file wrap_type_traits.h.

6.848.2 Member Typedef Documentation

6.848.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcpp::String >::wrap_category
```

Definition at line 85 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.849 Rcpp::traits::wrap_type_traits< short > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.849.1 Detailed Description

Definition at line 98 of file wrap_type_traits.h.

6.849.2 Member Typedef Documentation

6.849.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< short >::wrap_category
```

Definition at line 98 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.850 Rcpp::traits::wrap_type_traits< std::complex< double > > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.850.1 Detailed Description

Definition at line 91 of file wrap_type_traits.h.

6.850.2 Member Typedef Documentation

6.850.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::complex< double > >::wrap_category
```

Definition at line 91 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.851 Rcpp::traits::wrap_type_traits< std::complex< float > > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.851.1 Detailed Description

Definition at line 90 of file wrap_type_traits.h.

6.851.2 Member Typedef Documentation

6.851.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::complex< float > >::wrap_category
```

Definition at line 90 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.852 Rcpp::traits::wrap_type_traits< std::string > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.852.1 Detailed Description

Definition at line 83 of file wrap_type_traits.h.

6.852.2 Member Typedef Documentation

6.852.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::string >::wrap_category
```

Definition at line 83 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.853 Rcpp::traits::wrap_type_traits< std::wstring > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.853.1 Detailed Description

Definition at line 84 of file wrap_type_traits.h.

6.853.2 Member Typedef Documentation

6.853.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::wstring >::wrap_category
```

Definition at line 84 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.854 Rcpp::traits::wrap_type_traits< unsigned int > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.854.1 Detailed Description

Definition at line 81 of file wrap_type_traits.h.

6.854.2 Member Typedef Documentation

6.854.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< unsigned int >::wrap_category
```

Definition at line 81 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.855 Rcpp::traits::wrap_type_traits< unsigned long > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.855.1 Detailed Description

Definition at line 94 of file wrap_type_traits.h.

6.855.2 Member Typedef Documentation

6.855.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< unsigned long >::wrap_category
```

Definition at line 94 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.856 Rcpp::traits::wrap_type_traits< unsigned short > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.856.1 Detailed Description

Definition at line 99 of file wrap_type_traits.h.

6.856.2 Member Typedef Documentation

6.856.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< unsigned short >::wrap_category
```

Definition at line 99 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.857 Rcpp::traits::wrap_type_traits< wchar_t > Struct Reference

```
#include <wrap_type_traits.h>
```

Public Types

- typedef wrap_type_primitive_tag wrap_category

6.857.1 Detailed Description

Definition at line 87 of file wrap_type_traits.h.

6.857.2 Member Typedef Documentation

6.857.2.1 wrap_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< wchar_t >::wrap_category
```

Definition at line 87 of file wrap_type_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

6.858 Rcpp::traits::wrap_type_unknown_tag Struct Reference

```
#include <wrap_type_traits.h>
```

6.858.1 Detailed Description

unknown. Implicit conversion to SEXP will be tried.

Definition at line 37 of file wrap_type_traits.h.

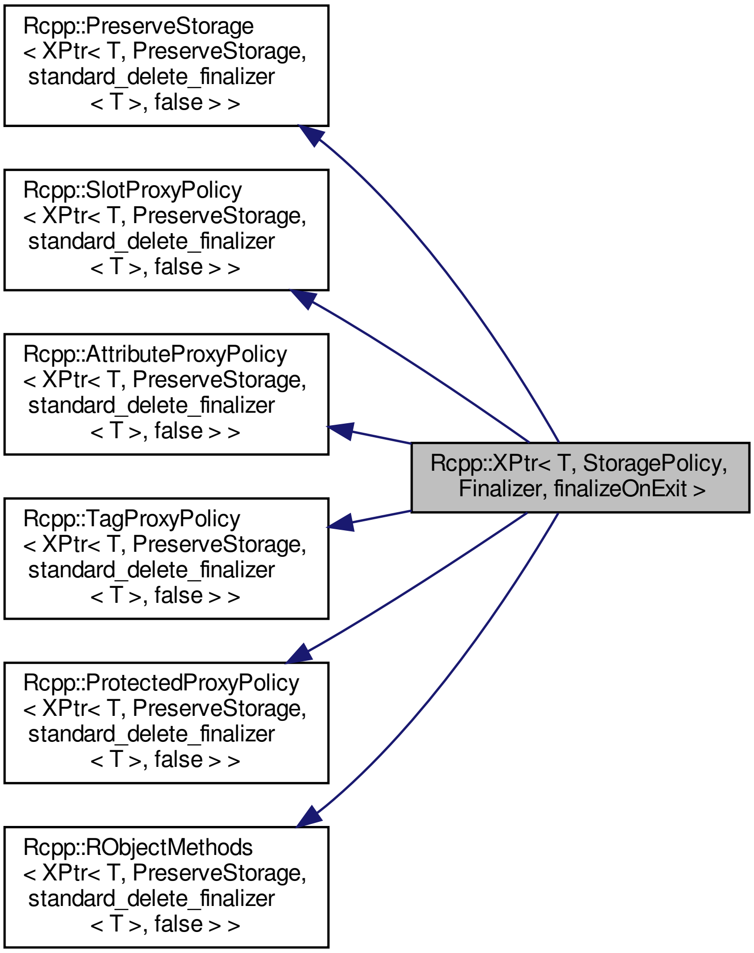
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap_type_traits.h

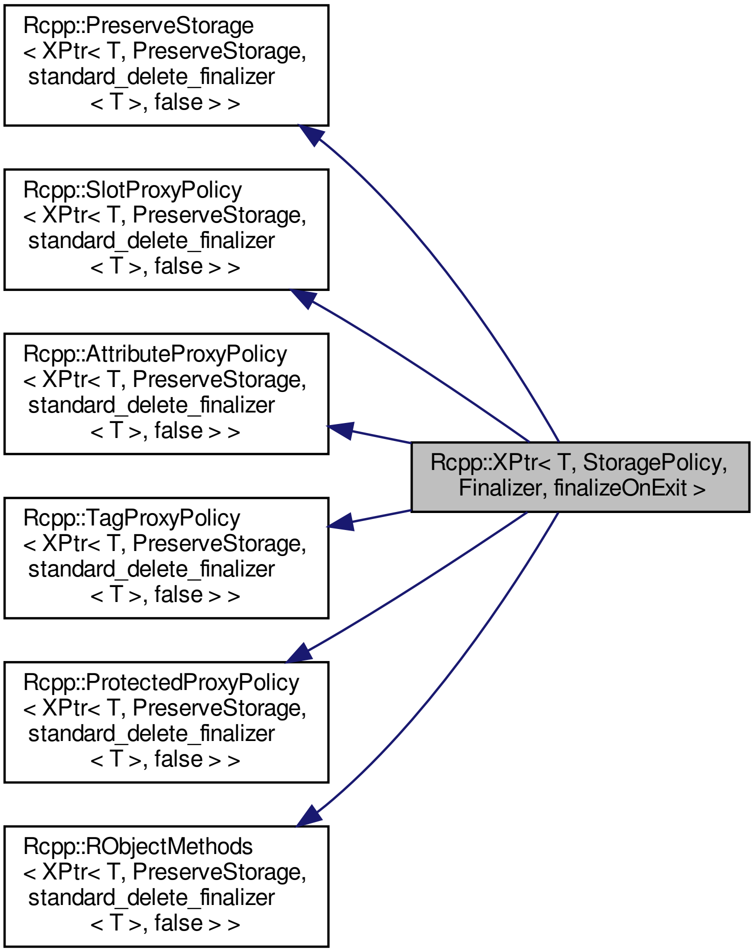
6.859 Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > Class Template Reference

```
#include <XPtr.h>
```

Inheritance diagram for Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >:


Rcpp::PreserveStorage < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::SlotProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::AttributeProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::TagProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::ProtectedProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::RObjectMethods < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >

Collaboration diagram for Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >:


Rcpp::PreserveStorage < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::SlotProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::AttributeProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::TagProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::ProtectedProxyPolicy < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::RObjectMethods < XPtr< T, PreserveStorage, standard_delete_finalizer < T >, false > >
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >

Public Types

- typedef StoragePolicy< XPtr > Storage
- typedef void(∗ unspecified_bool_type) ()

Public Member Functions

- XPtr (SEXP x)
- XPtr (SEXP x, SEXP tag, SEXP prot)
- XPtr (T ∗p, bool set_delete_finalizer=true, SEXP tag=R_NilValue, SEXP prot=R_NilValue)
- XPtr (const XPtr &other)
- XPtr & operator= (const XPtr &other)

- T ∗ get () const
- operator unspecified_bool_type () const
- bool operator! () const
- T ∗ checked_get () const
- T & operator∗ () const
- T ∗ operator-> () const
- void setDeleteFinalizer ()
- void release ()
- operator T∗ ()
- void update (SEXP)

Static Public Member Functions

- static void unspecified_bool_true ()

Private Member Functions

- void checked_set (SEXP x)

6.859.1 Detailed Description

template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>, bool finalizeOnExit = false>
class Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >

Definition at line 61 of file XPtr.h.

6.859.2 Member Typedef Documentation

6.859.2.1 Storage

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
typedef StoragePolicy<XPtr> Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::Storage
```

Definition at line 71 of file XPtr.h.

6.859.2.2 unspecified_bool_type

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
typedef void(* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_type)
()
```

Boolean operator wrapper for get() using the "safe bool idiom", see: http://www.boost.org/doc/libs/1_57_0/boost/smart_ptr/detail/operator_bool.hpp

Definition at line 134 of file XPtr.h.

6.859.3 Constructor & Destructor Documentation

6.859.3.1 XPtr() [1/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
              SEXP x )  [inline], [explicit]
```

constructs a XPtr wrapping the external pointer (EXTPTRSXP SEXP)

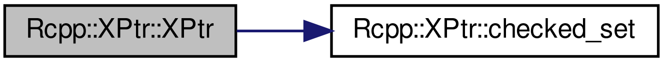
Parameters

xp	external pointer to wrap

Definition at line 78 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_set().

Here is the call graph for this function:



6.859.3.2 XPtr() [2/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
             SEXP x,
             SEXP tag,
             SEXP prot )  [inline], [explicit]
```

constructs a XPtr wrapping the external pointer (EXTPTRSXP SEXP)

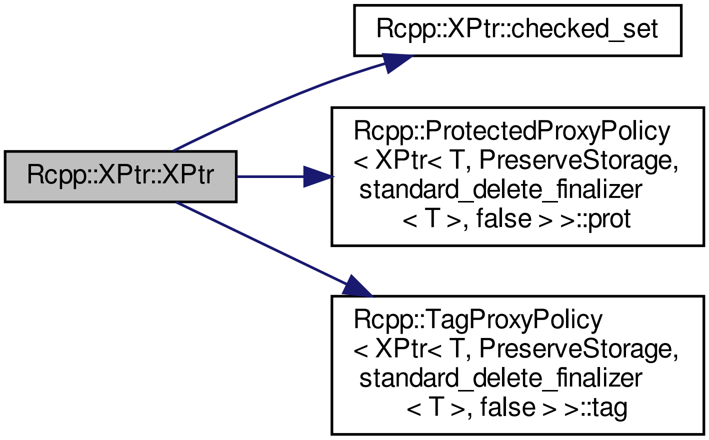
Parameters

xp	external pointer to wrap
tag	tag to assign to external pointer
prot	protected data to assign to external pointer

Definition at line 87 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_set(), Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::prot(), and Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::tag().

Here is the call graph for this function:


Rcpp::XPtr::XPtr
Rcpp::XPtr::checked_set
Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::prot
Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::tag

6.859.3.3 XPtr() [3/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
             T * p,
             bool set_delete_finalizer = true,
             SEXP tag = R_NilValue,
             SEXP prot = R_NilValue )  [inline], [explicit]
```

creates a new external pointer wrapping the dumb pointer p.

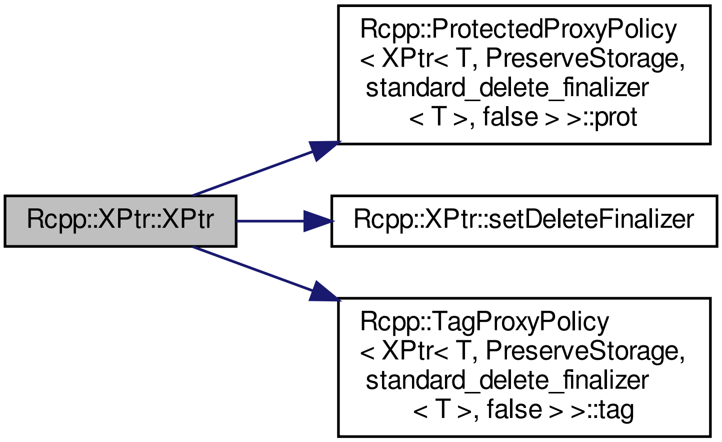
Parameters

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 104 of file XPtr.h.

References Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::prot(), RCPP_DEBUG_2, Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::setDeleteFinalizer(), and Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > >::tag().

Here is the call graph for this function:


Rcpp::ProtectedProxyPolicy
< XPtr< T, PreserveStorage,
standard_delete_finalizer
< T >, false > >::prot
Rcpp::XPtr::XPtr
Rcpp::XPtr::setDeleteFinalizer
Rcpp::TagProxyPolicy
< XPtr< T, PreserveStorage,
standard_delete_finalizer
< T >, false > >::tag

6.859.3.4 XPtr() [4/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
              const XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > & other )  [inline]
```

Definition at line 113 of file XPtr.h.

6.859.4 Member Function Documentation

6.859.4.1 checked_get()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get ( ) const  [inline]
```

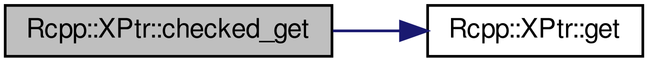
Access underlying pointer throwing an exception if the ptr is NULL

Definition at line 146 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get().

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator T∗(), Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator∗(), and Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator->().

Here is the call graph for this function:



6.859.4.2 checked_set()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_set (
              SEXP x )  [inline], [private]
```

Definition at line 202 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr().

6.859.4.3 get()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get ( ) const [inline]
```

Typesafe accessor for underlying pointer (use checked_get if you want an exception thrown if the pointer is NULL)

Definition at line 126 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get(), Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified_bool_type(), Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator!(), and Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::release().

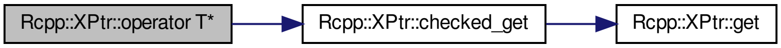
6.859.4.4 operator T∗()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator T* ( ) [inline]
```

Definition at line 195 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get().

Here is the call graph for this function:



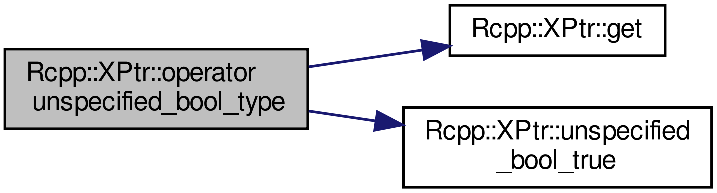
6.859.4.5 operator unspecified_bool_type()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified_bool_type ( )
const [inline]
```

Definition at line 136 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get(), and Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_true().

Here is the call graph for this function:


Rcpp::XPtr::operator unspecified_bool_type
Rcpp::XPtr::get
Rcpp::XPtr::unspecified _bool_true

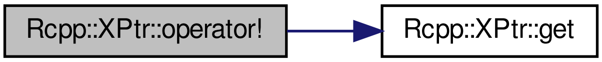
6.859.4.6 operator"!()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
bool Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator!  ( ) const  [inline]
```

Definition at line 139 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get().

Here is the call graph for this function:



6.859.4.7 operator∗()

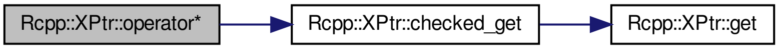
```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T& Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator* ( ) const  [inline]
```

Returns a reference to the object wrapped. This allows this object to look and feel like a dumb pointer to T

Definition at line 157 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get().

Here is the call graph for this function:



6.859.4.8 operator->()

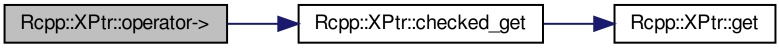
```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator-> ( ) const  [inline]
```

Returns the dumb pointer. This allows to call the -> operator on this as if it was the dumb pointer

Definition at line 165 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get().

Here is the call graph for this function:



6.859.4.9 operator=()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
XPtr& Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator= (
             const XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > & other )  [inline]
```

Definition at line 117 of file XPtr.h.

6.859.4.10 release()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::release ( )  [inline]
```

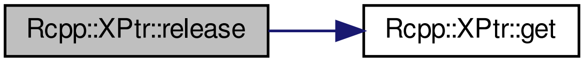
Release the external pointer (if any) immediately. This will cause the pointer to be deleted and it's storage to be set to NULL. After this call the get() method returns NULL and the checked_get() method throws an exception.

See the discussion here for the basic logic behind release: https://stat.ethz.ch/pipermail/r-help/2007-August/137871.html

Definition at line 183 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get().

Here is the call graph for this function:



6.859.4.11 setDeleteFinalizer()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::setDeleteFinalizer ( )  [inline]
```

Definition at line 169 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr().

6.859.4.12 unspecified_bool_true()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
static void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_true ( )
[inline], [static]
```

Definition at line 135 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified_bool_type().

6.859.4.13 update()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::update (
             SEXP   )  [inline]
```

Definition at line 199 of file XPtr.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/XPtr.h

6.860 Rcpp::traits::zero_type< T > Class Template Reference

```
#include <one_type.h>
```

Public Member Functions

- operator T () const

Private Member Functions

- Rcomplex op (true_type) const
- T op (false_type) const

6.860.1 Detailed Description

template<typename T>
class Rcpp::traits::zero_type< T >

Definition at line 55 of file one_type.h.

6.860.2 Member Function Documentation

6.860.2.1 op() [1/2]

```
template<typename T >
T Rcpp::traits::zero_type< T >::op (
             false_type  ) const  [inline], [private]
```

Definition at line 64 of file one_type.h.

6.860.2.2 op() [2/2]

```
template<typename T >
Rcomplex Rcpp::traits::zero_type< T >::op (
             true_type  ) const  [inline], [private]
```

Definition at line 57 of file one_type.h.

Referenced by Rcpp::traits::zero_type< T >::operator T().

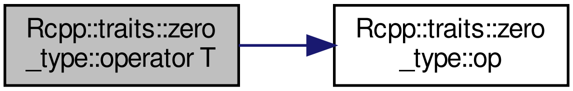
6.860.2.3 operator T()

```
template<typename T >
Rcpp::traits::zero_type< T >::operator T ( ) const  [inline]
```

Definition at line 69 of file one_type.h.

References Rcpp::traits::zero_type< T >::op().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/one_type.h

Chapter 7

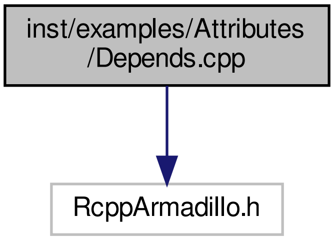
File Documentation

7.1 inst/examples/Attributes/cppFunction.R File Reference

7.2 inst/examples/Attributes/Depends.cpp File Reference

```
#include <RcppArmadillo.h>
```

Include dependency graph for Depends.cpp:



Functions

- List fastLm (NumericVector yr, NumericMatrix Xr)

7.2.1 Function Documentation

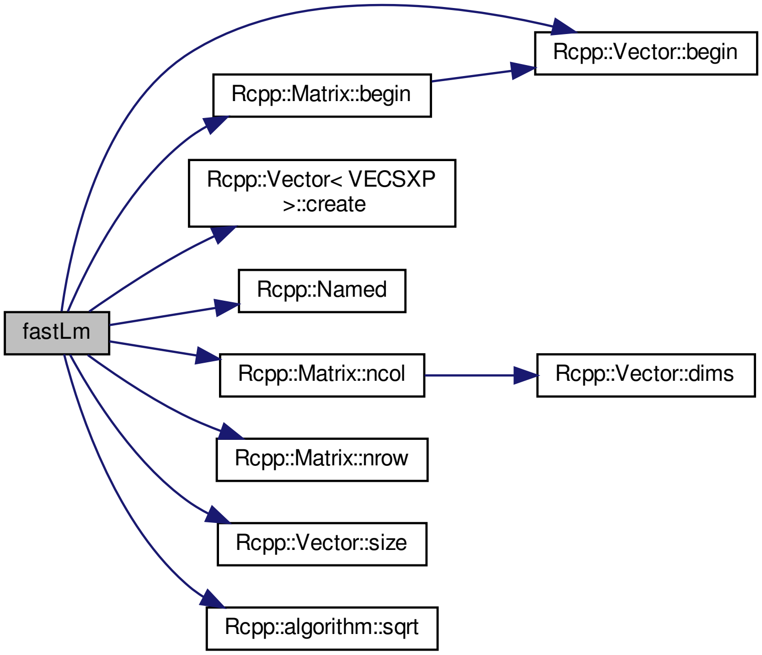
7.2.1.1 fastLm()

```
List fastLm (
            NumericVector yr,
            NumericMatrix Xr )
```

Definition at line 9 of file Depends.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< VECSXP >::create(), Rcpp::Named(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and Rcpp::algorithm::sqrt().

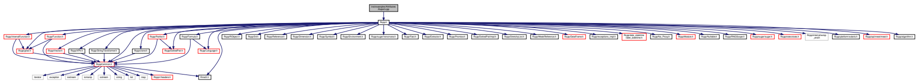
Here is the call graph for this function:



7.3 inst/examples/Attributes/Export.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for Export.cpp:



Functions

- int fibonacci (const int x)
- NumericVector convolve (NumericVector a, NumericVector b)
- List lapplyCpp (List input, Function f)

7.3.1 Function Documentation

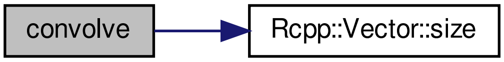
7.3.1.1 convolve()

```
NumericVector convolve (
             NumericVector a,
             NumericVector b )
```

Definition at line 17 of file Export.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



7.3.1.2 fibonacci()

```
int fibonacci (
             const int x )
```

Definition at line 7 of file Export.cpp.

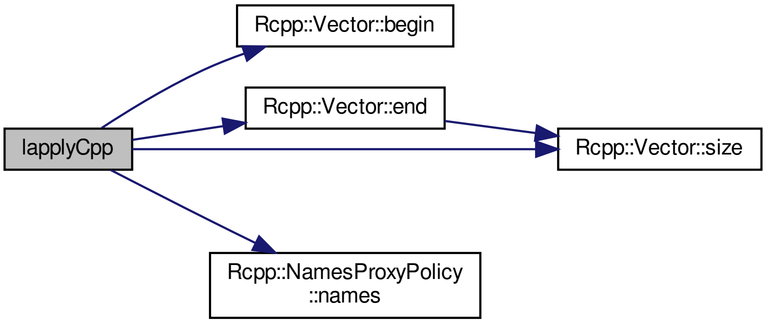
7.3.1.3 lapplyCpp()

```
List lapplyCpp (
             List input,
             Function f )
```

Definition at line 32 of file Export.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::NamesProxyPolicy< CLASS >::names(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

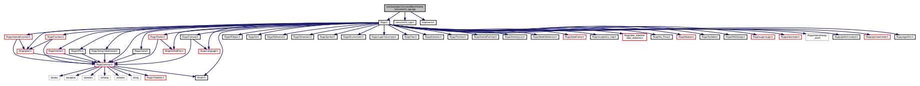


7.4 inst/examples/Attributes/sourceCpp.R File Reference

7.5 inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "convolve10_cpp.h"
#include "loopmacro.h"
```

Include dependency graph for convolve10_cpp.cpp:



Functions

- RcppExport SEXP convolve10cpp (SEXP a, SEXP b)

7.5.1 Function Documentation

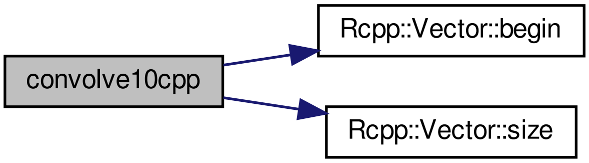
7.5.1.1 convolve10cpp()

```
RcppExport SEXP convolve10cpp (
              SEXP a,
              SEXP b )
```

Definition at line 17 of file convolve10_cpp.cpp.

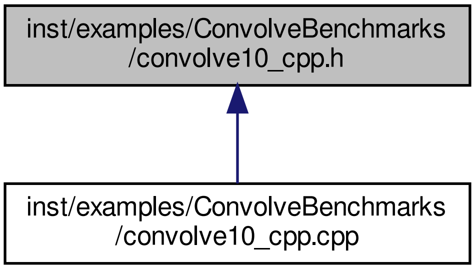
References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



7.6 inst/examples/ConvolveBenchmarks/convolve10_cpp.h File Reference

This graph shows which files directly or indirectly include this file:


inst/examples/ConvolveBenchmarks
/convolve10_cpp.h
inst/examples/ConvolveBenchmarks
/convolve10_cpp.cpp

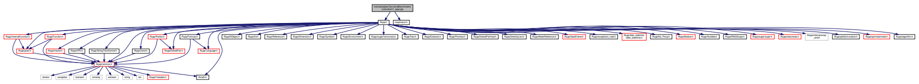
Classes

- class Cache
- class Vec

7.7 inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve11_cpp.cpp:



Functions

- RcppExport SEXP convolve11cpp (SEXP a, SEXP b)

7.7.1 Function Documentation

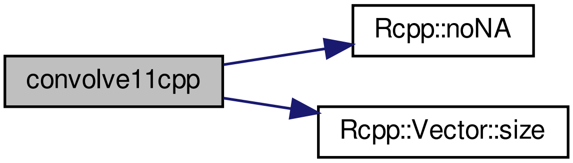
7.7.1.1 convolve11cpp()

```
RcppExport SEXP convolve11cpp (
            SEXP a,
            SEXP b )
```

Definition at line 10 of file convolve11_cpp.cpp.

References Rcpp::noNA(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

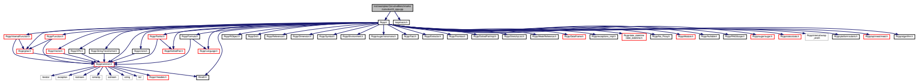
Here is the call graph for this function:



7.8 inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve12_cpp.cpp:



Functions

- RcppExport SEXP convolve12cpp (SEXP a, SEXP b)

7.8.1 Function Documentation

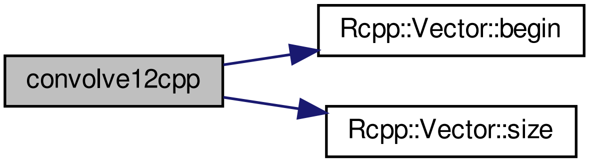
7.8.1.1 convolve12cpp()

```
RcppExport SEXP convolve12cpp (
             SEXP a,
             SEXP b )
```

Definition at line 7 of file convolve12_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

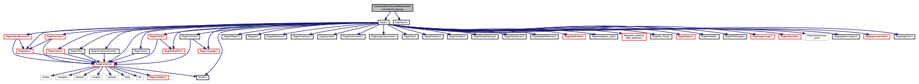
Here is the call graph for this function:



7.9 inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve13_cpp.cpp:



Functions

- template<typename T >
  T convolve (const T &a, const T &b)
- RcppExport SEXP convolve13cpp (SEXP a, SEXP b)

7.9.1 Function Documentation

7.9.1.1 convolve()

```
template<typename T >
T convolve (
             const T & a,
             const T & b )
```

Definition at line 8 of file convolve13_cpp.cpp.

Referenced by convolve13cpp().

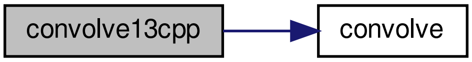
7.9.1.2 convolve13cpp()

```
RcppExport SEXP convolve13cpp (
             SEXP a,
             SEXP b )
```

Definition at line 21 of file convolve13_cpp.cpp.

References convolve().

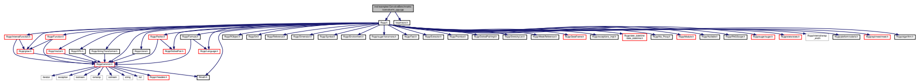
Here is the call graph for this function:



7.10 inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve14_cpp.cpp:



Functions

- RcppExport SEXP convolve14cpp (SEXP a, SEXP b)

7.10.1 Function Documentation

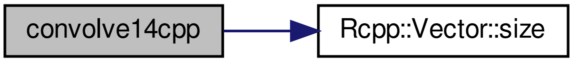
7.10.1.1 convolve14cpp()

```
RcppExport SEXP convolve14cpp (
              SEXP a,
              SEXP b )
```

Definition at line 8 of file convolve14_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

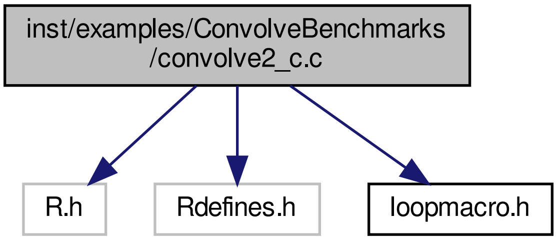
Here is the call graph for this function:



7.11 inst/examples/ConvolveBenchmarks/convolve2_c.c File Reference

```
#include <R.h>
#include <Rdefines.h>
#include "loopmacro.h"
```

Include dependency graph for convolve2_c.c:



Functions

- SEXP convolve2 (SEXP a, SEXP b)

7.11.1 Function Documentation

7.11.1.1 convolve2()

```
SEXP convolve2 (
            SEXP a,
            SEXP b )
```

Examples

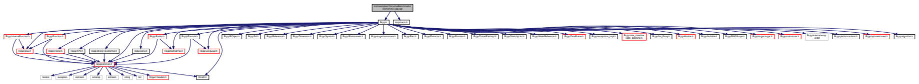
ConvolveBenchmarks/convolve2_c.c.

Definition at line 7 of file convolve2_c.c.

7.12 inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve3_cpp.cpp:



Functions

- RcppExport SEXP convolve3cpp (SEXP a, SEXP b)

7.12.1 Function Documentation

7.12.1.1 convolve3cpp()

```
RcppExport SEXP convolve3cpp (
            SEXP a,
            SEXP b )
```

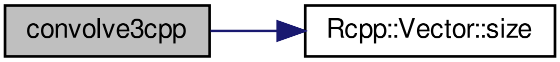
Examples

ConvolveBenchmarks/convolve3_cpp.cpp.

Definition at line 7 of file convolve3_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

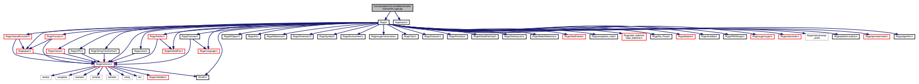
Here is the call graph for this function:



7.13 inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve4_cpp.cpp:



Functions

- RcppExport SEXP convolve4cpp (SEXP a, SEXP b)

7.13.1 Function Documentation

7.13.1.1 convolve4cpp()

```
RcppExport SEXP convolve4cpp (
            SEXP a,
            SEXP b )
```

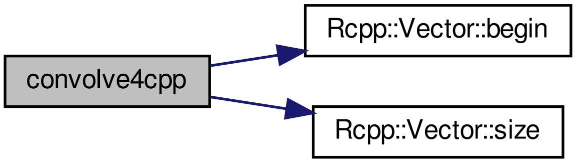
Examples

ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 7 of file convolve4_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

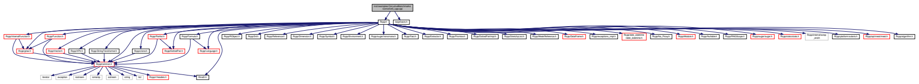
Here is the call graph for this function:



7.14 inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve5_cpp.cpp:



Functions

- RcppExport SEXP convolve5cpp (SEXP a, SEXP b)

7.14.1 Function Documentation

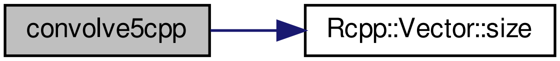
7.14.1.1 convolve5cpp()

```
RcppExport SEXP convolve5cpp (
            SEXP a,
            SEXP b )
```

Definition at line 9 of file convolve5_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

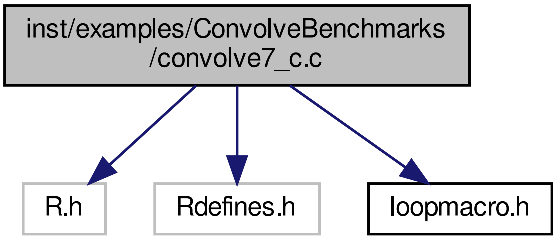
Here is the call graph for this function:



7.15 inst/examples/ConvolveBenchmarks/convolve7_c.c File Reference

```
#include <R.h>
#include <Rdefines.h>
#include "loopmacro.h"
```

Include dependency graph for convolve7_c.c:



Functions

- SEXP convolve7 (SEXP a, SEXP b)

7.15.1 Function Documentation

7.15.1.1 convolve7()

```
SEXP convolve7 (
            SEXP a,
            SEXP b )
```

Examples

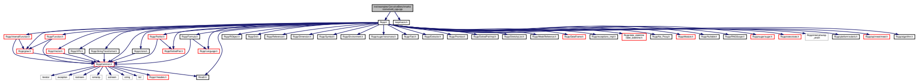
ConvolveBenchmarks/convolve7_c.c.

Definition at line 8 of file convolve7_c.c.

7.16 inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve8_cpp.cpp:



Classes

- class Vec

Functions

- RcppExport SEXP convolve8cpp (SEXP a, SEXP b)

7.16.1 Function Documentation

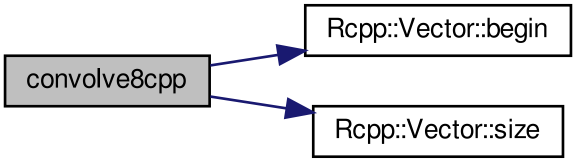
7.16.1.1 convolve8cpp()

```
RcppExport SEXP convolve8cpp (
             SEXP a,
             SEXP b )
```

Definition at line 24 of file convolve8_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

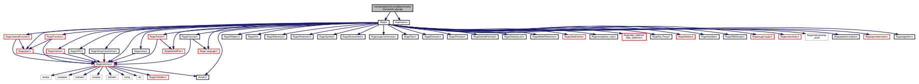
Here is the call graph for this function:



7.17 inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
```

Include dependency graph for convolve9_cpp.cpp:



Classes

- class Cache
- class Vec

Functions

- RcppExport SEXP convolve9cpp (SEXP a, SEXP b)

7.17.1 Function Documentation

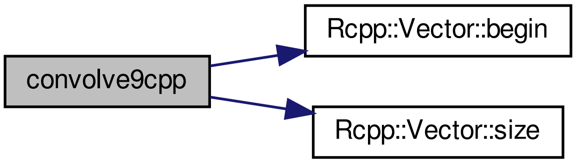
7.17.1.1 convolve9cpp()

```
RcppExport SEXP convolve9cpp (
             SEXP a,
             SEXP b )
```

Definition at line 42 of file convolve9_cpp.cpp.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

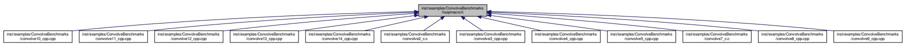
Here is the call graph for this function:



7.18 inst/examples/ConvolveBenchmarks/exampleRCode.r File Reference

7.19 inst/examples/ConvolveBenchmarks/loopmacro.h File Reference

This graph shows which files directly or indirectly include this file:


inst/examples/ConvolveBenchmarks/loopmacro.h
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve2_c.c
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve7_c.c
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp

Macros

- #define LOOPMACRO_C(name)
- #define LOOPMACRO_CPP(name) RcppExport LOOPMACRO_C(name)

7.19.1 Macro Definition Documentation

7.19.1.1 LOOPMACRO_C

```
#define LOOPMACRO_C(
                name )
```

Value:

```
SEXP name##__loop(SEXP n_, SEXP a, SEXP b){ \
    int n = INTEGER(n_)[0] ;                \
    SEXP res  = R_NilValue ;                \
    for( int i=0; i<n; i++){                \
       res = name( a, b ) ;                 \
    }                                       \
    return res ;                            \
}
```

Examples

ConvolveBenchmarks/convolve2_c.c, and ConvolveBenchmarks/convolve7_c.c.

Definition at line 2 of file loopmacro.h.

7.19.1.2 LOOPMACRO_CPP

```
#define LOOPMACRO_CPP(
                name ) RcppExport LOOPMACRO_C(name)
```

Examples

ConvolveBenchmarks/convolve3_cpp.cpp, and ConvolveBenchmarks/convolve4_cpp.cpp.

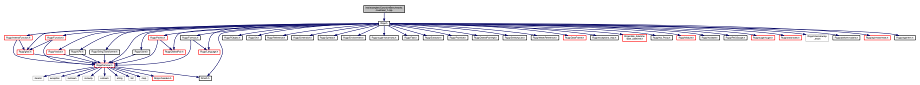
Definition at line 12 of file loopmacro.h.

7.20 inst/examples/ConvolveBenchmarks/overhead.r File Reference

7.21 inst/examples/ConvolveBenchmarks/overhead_1.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for overhead_1.cpp:



Functions

- SEXP overhead_cpp (SEXP a, SEXP b)
- void R_init_overhead_1 (DllInfo *info)

7.21.1 Function Documentation

7.21.1.1 overhead_cpp()

```
SEXP overhead_cpp (
            SEXP a,
            SEXP b )
```

Definition at line 8 of file overhead_1.cpp.

Referenced by R_init_overhead_1().

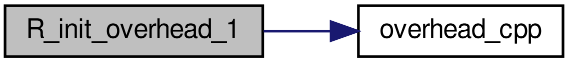
7.21.1.2 R_init_overhead_1()

```
void R_init_overhead_1 (
            DllInfo * info )
```

Definition at line 12 of file overhead_1.cpp.

References overhead_cpp().

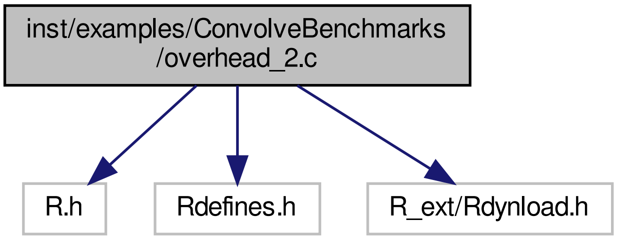
Here is the call graph for this function:



7.22 inst/examples/ConvolveBenchmarks/overhead_2.c File Reference

```
#include <R.h>
#include <Rdefines.h>
#include <R_ext/Rdynload.h>
```

Include dependency graph for overhead_2.c:



Functions

- SEXP overhead_c (SEXP a, SEXP b)
- void R_init_overhead_2 (DllInfo ∗info)

7.22.1 Function Documentation

7.22.1.1 overhead_c()

```
SEXP overhead_c (
            SEXP a,
            SEXP b )
```

Definition at line 8 of file overhead_2.c.

Referenced by R_init_overhead_2().

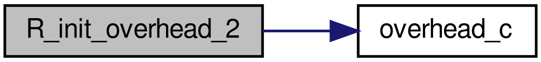
7.22.1.2 R_init_overhead_2()

```
void R_init_overhead_2 (
            DllInfo * info )
```

Definition at line 12 of file overhead_2.c.

References overhead_c().

Here is the call graph for this function:



7.23 inst/examples/FastLM/benchmark.r File Reference

7.24 inst/examples/FastLM/benchmarkLongley.r File Reference

7.25 inst/examples/FastLM/fastLMviaArmadillo.r File Reference

7.26 inst/examples/FastLM/fastLMviaGSL.r File Reference

7.27 inst/examples/FastLM/lmArmadillo.R File Reference

7.28 inst/examples/FastLM/lmGSL.R File Reference

7.29 inst/examples/functionCallback/newApiExample.r File Reference

7.30 inst/examples/Misc/fibonacci.r File Reference

7.31 inst/examples/Misc/ifelseLooped.r File Reference

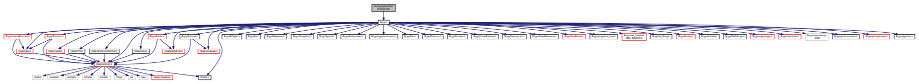
7.32 inst/examples/Misc/newFib.r File Reference

7.33 inst/examples/Misc/piBySimulation.r File Reference

7.34 inst/examples/Misc/piSugar.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for piSugar.cpp:



Functions

- double piSugar (const int N)

7.34.1 Function Documentation

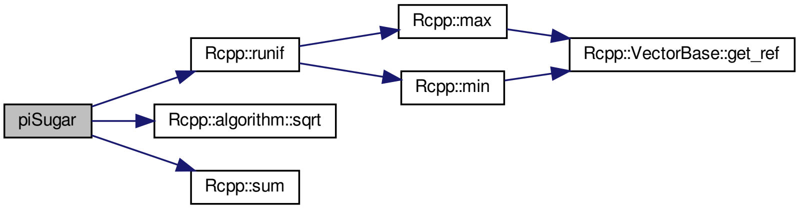
7.34.1.1 piSugar()

```
double piSugar (
             const int N )
```

Definition at line 7 of file piSugar.cpp.

References Rcpp::runif(), Rcpp::algorithm::sqrt(), and Rcpp::sum().

Here is the call graph for this function:



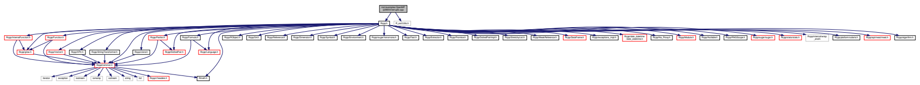
7.35 inst/examples/OpenMP/check.R File Reference

7.36 inst/examples/OpenMP/OpenMPandInline.r File Reference

7.37 inst/examples/OpenMP/piWithInterrupts.cpp File Reference

```
#include <Rcpp.h>
#include <R_ext/Utils.h>
```

Include dependency graph for piWithInterrupts.cpp:



Classes

- class interrupt_exception

Functions

- static void check_interrupt_impl (void ∗)
- bool check_interrupt ()
- RcppExport SEXP PiLeibniz (SEXP n, SEXP frequency)

7.37.1 Function Documentation

7.37.1.1 check_interrupt()

```
bool check_interrupt ( )  [inline]
```

Call this method to check for user interrupts. This is based on the results of a discussion on the R-devel mailing list, suggested by Simon Urbanek.

Attention

> This method must not be called by any other thread than the master thread. If called from within an OpenMP parallel for loop, make sure to check for omp_get_thread_num()==0 before calling this method!

Returns

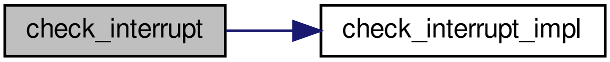
True, if a user interrupt has been detected.

Definition at line 64 of file piWithInterrupts.cpp.

References check_interrupt_impl().

Referenced by PiLeibniz().

Here is the call graph for this function:



7.37.1.2 check_interrupt_impl()

```
static void check_interrupt_impl (
            void * )  [inline], [static]
```

Do the actual check for an interrupt.

Attention

This method should never be called directly.

Parameters

in	*dummy*	Dummy argument.

Definition at line 50 of file piWithInterrupts.cpp.

Referenced by check_interrupt().

7.37.1.3 PiLeibniz()

```
RcppExport SEXP PiLeibniz (
              SEXP n,
              SEXP frequency )
```

Compute pi using the Leibniz formula (a very inefficient approach).

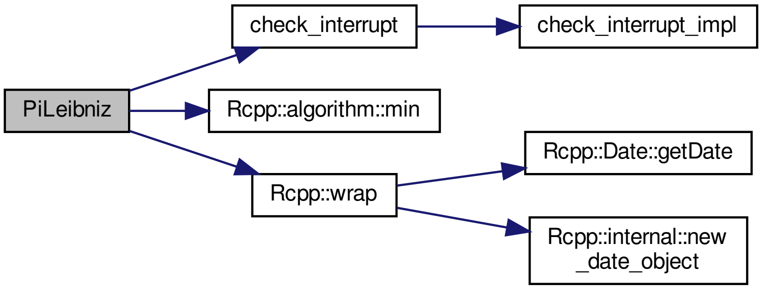
Parameters

in	*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, Rcpp::algorithm::min(), and Rcpp::wrap().

Here is the call graph for this function:



7.38 inst/examples/performance/extractors.R File Reference

7.39 inst/examples/performance/performance.R File Reference

7.40 inst/examples/RcppGibbs/RcppGibbs.R File Reference

7.41 inst/examples/RcppGibbs/RcppGibbs_Updated.R File Reference

7.42 inst/examples/RcppGibbs/timeRNGs.R File Reference

7.43 inst/examples/RcppInline/external_pointer.r File Reference

7.44 inst/examples/RcppInline/RcppInlineExample.r File Reference

7.45 inst/examples/RcppInline/RcppInlineWithLibsExamples.r File Reference

7.46 inst/examples/RcppInline/RcppSimpleExample.r File Reference

7.47 inst/examples/RcppInline/RObject.r File Reference

7.48 inst/examples/RcppInline/UncaughtExceptions.r File Reference

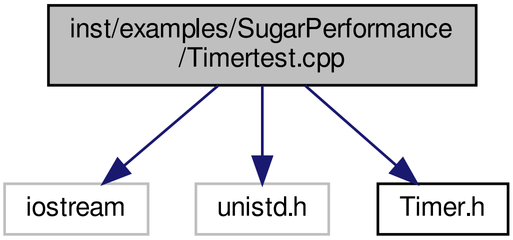
7.49 inst/examples/SugarPerformance/sugarBenchmarks.R File Reference

7.50 inst/examples/SugarPerformance/Timertest.cpp File Reference

```
#include <iostream>
#include <unistd.h>
```

```
#include "Timer.h"
```

Include dependency graph for Timertest.cpp:



Functions

- int main ()

7.50.1 Function Documentation

7.50.1.1 main()

```
int main ( )
```

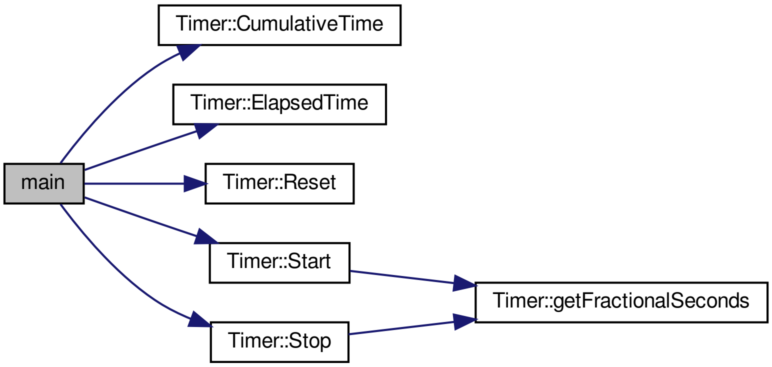
Examples

SugarPerformance/Timertest.cpp.

Definition at line 9 of file Timertest.cpp.

References Timer::CumulativeTime(), Timer::ElapsedTime(), Timer::Reset(), Timer::Start(), and Timer::Stop().

Here is the call graph for this function:



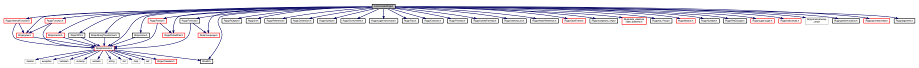
7.51 inst/include/doxygen/Examples.h File Reference

7.52 inst/include/Rcpp.h File Reference

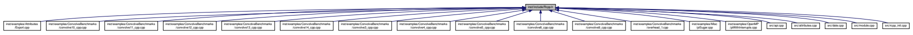
```
#include <RcppCommon.h>
#include <Rcpp/RObject.h>
#include <Rcpp/S4.h>
#include <Rcpp/Reference.h>
#include <Rcpp/clone.h>
#include <Rcpp/grow.h>
#include <Rcpp/Dimension.h>
#include <Rcpp/Symbol.h>
#include <Rcpp/Environment.h>
#include <Rcpp/Vector.h>
#include <Rcpp/sugar/nona/nona.h>
#include <Rcpp/Fast.h>
#include <Rcpp/Extractor.h>
#include <Rcpp/Promise.h>
#include <Rcpp/XPtr.h>
#include <Rcpp/DottedPairImpl.h>
#include <Rcpp/Function.h>
#include <Rcpp/Language.h>
#include <Rcpp/DottedPair.h>
#include <Rcpp/Pairlist.h>
#include <Rcpp/StretchyList.h>
#include <Rcpp/WeakReference.h>
#include <Rcpp/StringTransformer.h>
#include <Rcpp/Formula.h>
#include <Rcpp/DataFrame.h>
#include <Rcpp/exceptions_impl.h>
```

```
#include <Rcpp/date_datetime/date_datetime.h>
#include <Rcpp/Na_Proxy.h>
#include <Rcpp/Module.h>
#include <Rcpp/InternalFunction.h>
#include <Rcpp/Nullable.h>
#include <Rcpp/RNGScope.h>
#include <Rcpp/sugar/sugar.h>
#include <Rcpp/stats/stats.h>
#include <Rcpp/Rmath.h>
#include <Rcpp/internal/wrap_end.h>
#include <Rcpp/platform/solaris.h>
#include <Rcpp/api/meat/meat.h>
#include <Rcpp/algorithm.h>
```

Include dependency graph for Rcpp.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_NEW_DATE_DATETIME_VECTORS 1

7.52.1 Macro Definition Documentation

7.52.1.1 RCPP_NEW_DATE_DATETIME_VECTORS

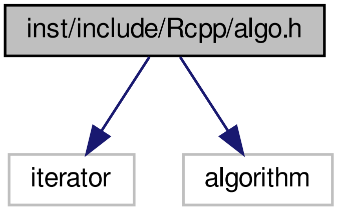
```
#define RCPP_NEW_DATE_DATETIME_VECTORS 1
```

Definition at line 62 of file Rcpp.h.

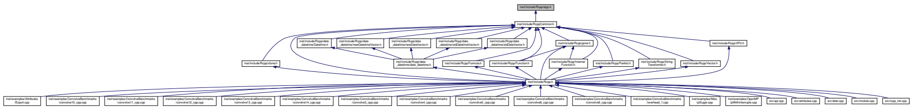
7.53 inst/include/Rcpp/algo.h File Reference

```
#include <iterator>
#include <algorithm>
```

Include dependency graph for algo.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

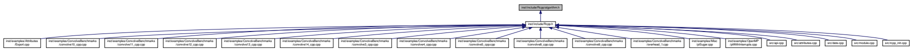
  Rcpp API.

Functions

- template<class InputIterator , class T >
  bool Rcpp::__any (InputIterator first, InputIterator last, const T &value, std::input_iterator_tag)
- template<class RandomAccessIterator , class T >
  bool Rcpp::__any (RandomAccessIterator __first, RandomAccessIterator __last, const T &__val, std::random_access_iterator_tag)
- template<class InputIterator , class T >
  bool Rcpp::any (InputIterator first, InputIterator last, const T &value)
- template<class InputIterator , class Predicate >
  bool Rcpp::__any_if (InputIterator first, InputIterator last, Predicate pred, std::input_iterator_tag)
- template<class RandomAccessIterator , class Predicate >
  bool Rcpp::__any_if (RandomAccessIterator __first, RandomAccessIterator __last, Predicate __pred, std::random_access_iterator_tag)
- template<class InputIterator , class Predicate >
  bool Rcpp::any_if (InputIterator first, InputIterator last, Predicate pred)

7.54 inst/include/Rcpp/algorithm.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/algorithm.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::algorithm::helpers::CTYPE_CHAR
- struct Rcpp::algorithm::helpers::CTYPE_SHORT
- struct Rcpp::algorithm::helpers::CTYPE_INT
- struct Rcpp::algorithm::helpers::CTYPE_LONG
- struct Rcpp::algorithm::helpers::CTYPE_FLOAT
- struct Rcpp::algorithm::helpers::CTYPE_DOUBLE
- struct Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE
- struct Rcpp::algorithm::helpers::CTYPE_STRING
- struct Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR
- struct Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT
- struct Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT
- struct Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG
- struct Rcpp::algorithm::helpers::CTYPE_UNKNOWN
- struct Rcpp::algorithm::helpers::ctype_helper< I >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >
- struct Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >
- struct Rcpp::algorithm::helpers::ctype< T >
- struct Rcpp::algorithm::helpers::decays_to_ctype< T >
- struct Rcpp::algorithm::helpers::rtype_helper< T >
- struct Rcpp::algorithm::helpers::rtype_helper< double >
- struct Rcpp::algorithm::helpers::rtype_helper< int >
- struct Rcpp::algorithm::helpers::rtype< T >
- struct Rcpp::algorithm::helpers::log
- struct Rcpp::algorithm::helpers::exp
- struct Rcpp::algorithm::helpers::sqrt

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::algorithm
- Rcpp::algorithm::helpers

Macros

- #define RCPP_CONSTEXPR_FUNC
- #define RCPP_CONSTEXPR_VAR const

Functions

- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::sum (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::sum_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::prod (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::prod_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::max (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::max_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::min (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type Rcpp::algorithm::min_nona (InputIterator begin, InputIterator end)
- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value &&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::type, double >::value, double >::type Rcpp::algorithm::mean (InputIterator begin, InputIterator end)

- template<typename InputIterator >
  traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value &&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↩ ::type, int >::value, double >::type Rcpp::algorithm::mean (InputIterator begin, InputIterator end)
- template<typename InputIterator , typename OutputIterator >
  void Rcpp::algorithm::log (InputIterator begin, InputIterator end, OutputIterator out)
- template<typename InputIterator , typename OutputIterator >
  void Rcpp::algorithm::exp (InputIterator begin, InputIterator end, OutputIterator out)
- template<typename InputIterator , typename OutputIterator >
  void Rcpp::algorithm::sqrt (InputIterator begin, InputIterator end, OutputIterator out)

7.54.1 Macro Definition Documentation

7.54.1.1 RCPP_CONSTEXPR_FUNC

```
#define RCPP_CONSTEXPR_FUNC
```

Definition at line 29 of file algorithm.h.

7.54.1.2 RCPP_CONSTEXPR_VAR

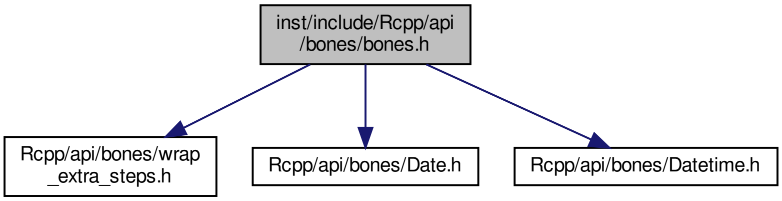
```
#define RCPP_CONSTEXPR_VAR const
```

Definition at line 30 of file algorithm.h.

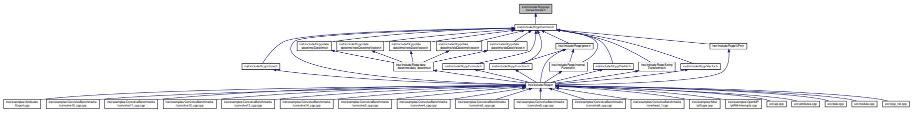
7.55 inst/include/Rcpp/api/bones/bones.h File Reference

```
#include <Rcpp/api/bones/wrap_extra_steps.h>
#include <Rcpp/api/bones/Date.h>
#include <Rcpp/api/bones/Datetime.h>
```

Include dependency graph for bones.h:

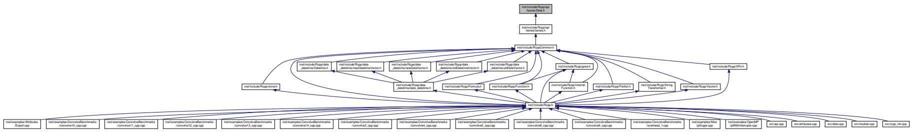


This graph shows which files directly or indirectly include this file:



7.56 inst/include/Rcpp/api/bones/Date.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::wrap_type_traits< Rcpp::Date >
- struct Rcpp::traits::r_type_traits< Rcpp::Date >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > >
- struct Rcpp::traits::r_sexptype_traits< Rcpp::Date >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap
- Rcpp::internal

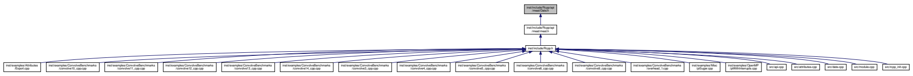
  internal implementation details

Functions

- template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Date > (SEXP)
- template<> double Rcpp::internal::caster< Rcpp::Date, double > (Rcpp::Date from)
- template<> Rcpp::Date Rcpp::internal::caster< double, Rcpp::Date > (double from)

7.57 inst/include/Rcpp/api/meat/Date.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

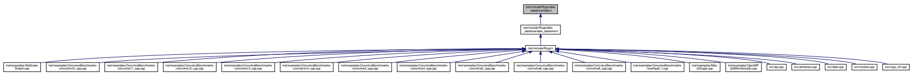
  Rcpp API.

Functions

- template<> SEXP Rcpp::wrap (const Date &date)

7.58 inst/include/Rcpp/date_datetime/Date.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Date

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

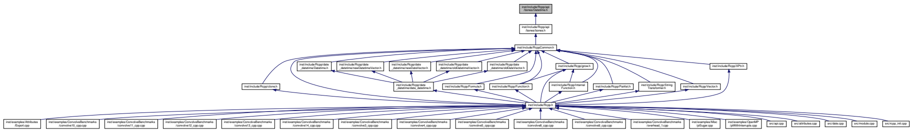
  internal implementation details

Functions

- template<> SEXP Rcpp::wrap< Rcpp::Date > (const Rcpp::Date &date)
- template<> double Rcpp::internal::caster< Rcpp::Date, double > (Rcpp::Date from)
- template<> Rcpp::Date Rcpp::internal::caster< double, Rcpp::Date > (double from)
- template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Date > (SEXP)
- Date Rcpp::operator+ (const Date &date, int offset)
- double Rcpp::operator- (const Date &d1, const Date &d2)
- bool Rcpp::operator< (const Date &d1, const Date &d2)
- bool Rcpp::operator> (const Date &d1, const Date &d2)
- bool Rcpp::operator== (const Date &d1, const Date &d2)
- bool Rcpp::operator>= (const Date &d1, const Date &d2)
- bool Rcpp::operator<= (const Date &d1, const Date &d2)
- bool Rcpp::operator!= (const Date &d1, const Date &d2)
- std::ostream & Rcpp::operator<< (std::ostream &os, const Date d)
- SEXP Rcpp::internal::getPosixClasses ()
- SEXP Rcpp::internal::new_posixt_object (double d)
- SEXP Rcpp::internal::new_date_object (double d)

7.59 inst/include/Rcpp/api/bones/Datetime.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::wrap_type_traits< Rcpp::Datetime >
- struct Rcpp::traits::r_type_traits< Rcpp::Datetime >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > >
- struct Rcpp::traits::r_sexptype_traits< Rcpp::Datetime >

Namespaces

- Rcpp

    Rcpp API.
- Rcpp::traits

    traits used to dispatch wrap
- Rcpp::internal

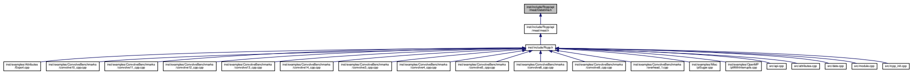
    internal implementation details

Functions

- template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)
- template<> double Rcpp::internal::caster< Rcpp::Datetime, double > (Rcpp::Datetime from)
- template<> Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (double from)

7.60 inst/include/Rcpp/api/meat/Datetime.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

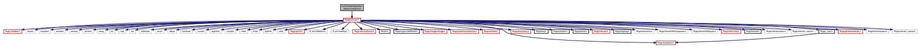
Functions

- template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)
- template<> SEXP Rcpp::wrap< Datetime > (const Datetime &date)

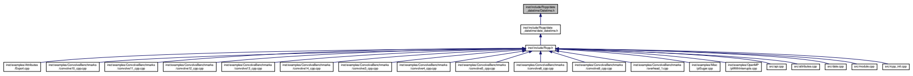
7.61 inst/include/Rcpp/date_datetime/Datetime.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for Datetime.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Datetime

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

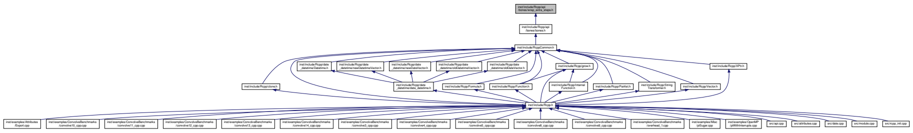
  internal implementation details

Functions

- template<> SEXP Rcpp::wrap< Rcpp::Datetime > (const Rcpp::Datetime &dt)
- template<> double Rcpp::internal::caster< Rcpp::Datetime, double > (Rcpp::Datetime from)
- template<> Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (double from)
- template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)
- Datetime Rcpp::operator+ (const Datetime &datetime, double offset)
- Datetime Rcpp::operator+ (const Datetime &datetime, int offset)
- double Rcpp::operator- (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator< (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator> (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator== (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator>= (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator<= (const Datetime &d1, const Datetime &d2)
- bool Rcpp::operator!= (const Datetime &d1, const Datetime &d2)
- std::ostream & Rcpp::operator<< (std::ostream &os, const Datetime d)

7.62 inst/include/Rcpp/api/bones/wrap_extra_steps.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/bones/wrap_extra_steps.h
inst/include/Rcpp/api/bones/bones.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

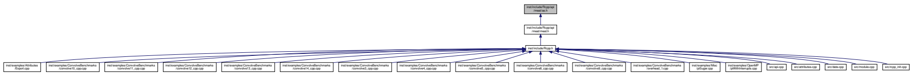
  Rcpp API.

Functions

- template<typename T >
  SEXP Rcpp::wrap_extra_steps (SEXP x)

7.63 inst/include/Rcpp/api/meat/as.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

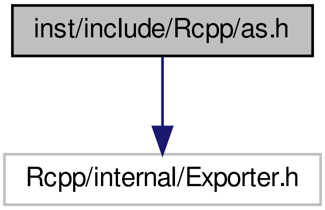
Functions

- void ∗ Rcpp::internal::as_module_object_internal (SEXP obj)

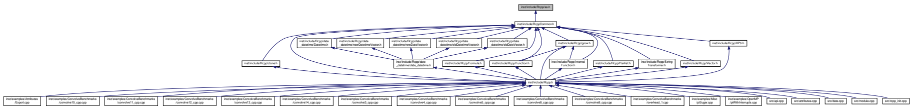
7.64 inst/include/Rcpp/as.h File Reference

```
#include <Rcpp/internal/Exporter.h>
```

Include dependency graph for as.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

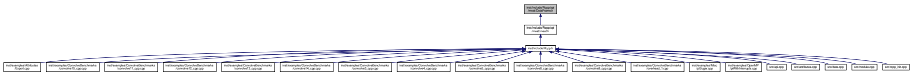
  internal implementation details

Functions

- template<typename T >
  T Rcpp::internal::primitive_as (SEXP x)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_primitive_tag)
- const char ∗ Rcpp::internal::check_single_string (SEXP x)
- template<typename T >
  T Rcpp::internal::as_string (SEXP x, Rcpp::traits::true_type)
- template<typename T >
  T Rcpp::internal::as_string (SEXP x, Rcpp::traits::false_type)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_string_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_RcppString_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_generic_tag)
- void ∗ Rcpp::internal::as_module_object_internal (SEXP obj)
- template<typename T >
  object< T > Rcpp::internal::as_module_object (SEXP x)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_const_pointer_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_pointer_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_reference_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_const_reference_tag)
- template<typename T >
  T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_enum_tag)
- template<typename T >
  T Rcpp::as (SEXP x)
- template<> char Rcpp::as< char > (SEXP x)
- template<typename T >
  traits::remove_const_and_reference< T >::type Rcpp::bare_as (SEXP x)
- template<> SEXP Rcpp::as (SEXP x)

7.65 inst/include/Rcpp/api/meat/DataFrame.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

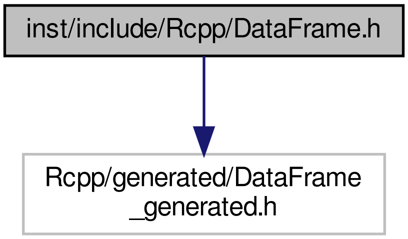
- Rcpp

  Rcpp API.

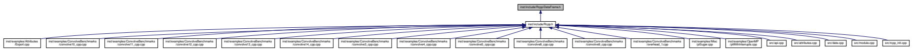
7.66 inst/include/Rcpp/DataFrame.h File Reference

```
#include <Rcpp/generated/DataFrame_generated.h>
```

Include dependency graph for DataFrame.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DataFrame_Impl< StoragePolicy >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Typedefs

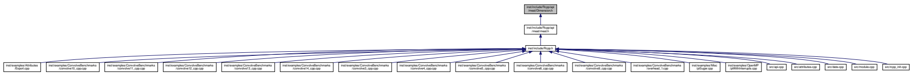
- typedef DataFrame_Impl< PreserveStorage > Rcpp::DataFrame

Functions

- SEXP Rcpp::internal::empty_data_frame ()

7.67 inst/include/Rcpp/api/meat/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:



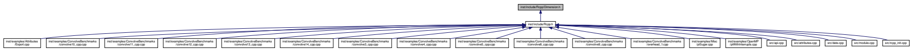
Namespaces

- Rcpp

  Rcpp API.

7.68 inst/include/Rcpp/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Dimension

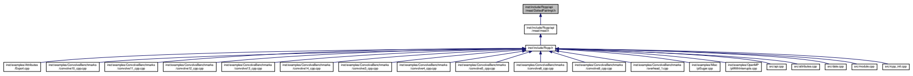
Namespaces

- [Rcpp](#)

    Rcpp API.

7.69 inst/include/Rcpp/api/meat/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/DottedPairImpl.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

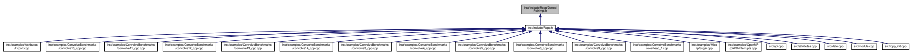
Namespaces

- [Rcpp](#)

    Rcpp API.

7.70 inst/include/Rcpp/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DottedPairImpl< CLASS >

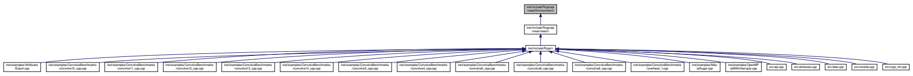
Namespaces

- [Rcpp](#)

    Rcpp API.

7.71 inst/include/Rcpp/api/meat/Environment.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

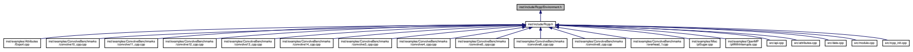
  Rcpp API.

Functions

- Environment Rcpp::new_env (int size=29)
- Environment Rcpp::new_env (SEXP parent, int size=29)

7.72 inst/include/Rcpp/Environment.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef Environment_Impl< PreserveStorage > Rcpp::Environment

Functions

- Rcpp::RCPP_API_CLASS (Environment_Impl)
- Rcpp::Environment_Impl (SEXP x)
- Rcpp::Environment_Impl (const std::string &name)
- Rcpp::Environment_Impl (int pos)
- SEXP Rcpp::ls (bool all) const
- SEXP Rcpp::get (const std::string &name) const
- SEXP Rcpp::get (Symbol name) const
- SEXP Rcpp::find (const std::string &name) const
- SEXP Rcpp::find (Symbol name) const
- bool Rcpp::exists (const std::string &name) const
- bool Rcpp::assign (const std::string &name, SEXP x) const
- bool Rcpp::assign (const std::string &name, const Shield< SEXP > &x) const
- template<typename WRAPPABLE >
  bool Rcpp::assign (const std::string &name, const WRAPPABLE &x) const
- bool Rcpp::isLocked () const

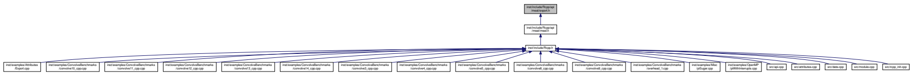
- bool Rcpp::remove (const std::string &name)
- void Rcpp::lock (bool bindings=false)
- void Rcpp::lockBinding (const std::string &name)
- void Rcpp::unlockBinding (const std::string &name)
- bool Rcpp::bindingIsLocked (const std::string &name) const
- bool Rcpp::bindingIsActive (const std::string &name) 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) const
- void Rcpp::update (SEXP)

Variables

- public Rcpp::BindingPolicy< Environment_Impl< StoragePolicy > >

7.73 inst/include/Rcpp/api/meat/export.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/export.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::traits::ContainerExporter< ContainerTemplate, T >
- struct Rcpp::traits::container_exporter< Container, int >
- struct Rcpp::traits::container_exporter< Container, double >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::traits

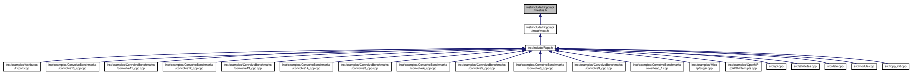
  traits used to dispatch wrap

Functions

- template<typename InputIterator , typename value_type >
  void Rcpp::internal::export_range__dispatch (SEXP x, InputIterator first, ::Rcpp::traits::r_type_generic_tag)

7.74 inst/include/Rcpp/api/meat/is.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/is.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

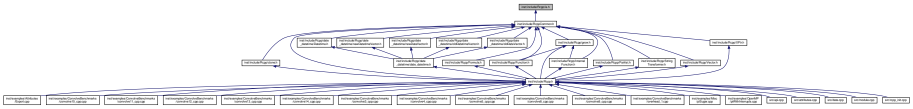
Functions

- bool Rcpp::internal::is_atomic (SEXP x)
- bool Rcpp::internal::is_matrix (SEXP x)
- template<> bool Rcpp::internal::is__simple< int > (SEXP x)
- template<> bool Rcpp::internal::is__simple< double > (SEXP x)
- template<> bool Rcpp::internal::is__simple< bool > (SEXP x)
- template<> bool Rcpp::internal::is__simple< std::string > (SEXP x)
- template<> bool Rcpp::internal::is__simple< String > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Rcomplex > (SEXP x)
- template<> bool Rcpp::internal::is__simple< CharacterVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< CharacterMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< RObject > (SEXP)
- template<> bool Rcpp::internal::is__simple< IntegerVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< ComplexVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< RawVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< NumericVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< LogicalVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Language > (SEXP x)
- template<> bool Rcpp::internal::is__simple< DottedPair > (SEXP x)
- template<> bool Rcpp::internal::is__simple< List > (SEXP x)
- template<> bool Rcpp::internal::is__simple< IntegerMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< ComplexMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< RawMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< NumericMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< LogicalMatrix > (SEXP x)

- template<> bool Rcpp::internal::is__simple< GenericMatrix > (SEXP x)
- template<> bool Rcpp::internal::is__simple< DataFrame > (SEXP x)
- template<> bool Rcpp::internal::is__simple< WeakReference > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Symbol > (SEXP x)
- template<> bool Rcpp::internal::is__simple< S4 > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Reference > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Promise > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Pairlist > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Function > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Environment > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Formula > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Date > (SEXP x)
- template<> bool Rcpp::internal::is__simple< Datetime > (SEXP x)
- template<> bool Rcpp::internal::is__simple< DateVector > (SEXP x)
- template<> bool Rcpp::internal::is__simple< DatetimeVector > (SEXP x)
- bool Rcpp::internal::is_module_object_internal (SEXP obj, const char ∗clazz)
- template<typename T >
  bool Rcpp::internal::is__module__object (SEXP x)

7.75 inst/include/Rcpp/is.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

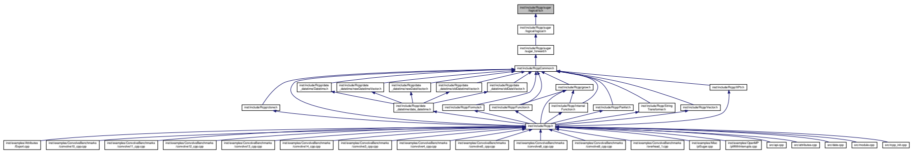
  internal implementation details

Functions

- template<typename T >
  bool Rcpp::internal::is__simple (SEXP x)
- template<typename T >
  bool Rcpp::internal::is__module__object (SEXP x)
- template<typename T >
  bool Rcpp::internal::is__dispatch (SEXP x, Rcpp::traits::false_type)
- template<typename T >
  bool Rcpp::internal::is__dispatch (SEXP x, Rcpp::traits::true_type)
- template<typename T >
  bool Rcpp::is (SEXP x)

7.76 inst/include/Rcpp/sugar/logical/is.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.

Functions

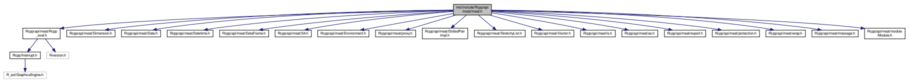
- template<bool NA, typename T >
  bool Rcpp::is_true (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool Rcpp::is_false (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)
- template<bool NA, typename T >
  bool Rcpp::is_na (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)

7.77 inst/include/Rcpp/api/meat/meat.h File Reference

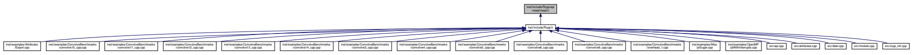
```
#include <Rcpp/api/meat/Rcpp_eval.h>
#include <Rcpp/api/meat/Dimension.h>
#include <Rcpp/api/meat/Date.h>
#include <Rcpp/api/meat/Datetime.h>
#include <Rcpp/api/meat/DataFrame.h>
#include <Rcpp/api/meat/S4.h>
#include <Rcpp/api/meat/Environment.h>
#include <Rcpp/api/meat/proxy.h>
#include <Rcpp/api/meat/DottedPairImpl.h>
#include <Rcpp/api/meat/StretchyList.h>
#include <Rcpp/api/meat/Vector.h>
#include <Rcpp/api/meat/is.h>
#include <Rcpp/api/meat/as.h>
#include <Rcpp/api/meat/export.h>
#include <Rcpp/api/meat/protection.h>
#include <Rcpp/api/meat/wrap.h>
#include <Rcpp/api/meat/message.h>
```

```
#include <Rcpp/api/meat/module/Module.h>
```

Include dependency graph for meat.h:

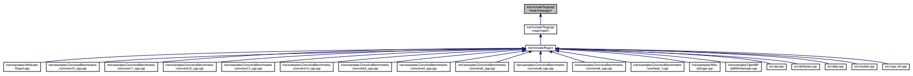


This graph shows which files directly or indirectly include this file:



7.78 inst/include/Rcpp/api/meat/message.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

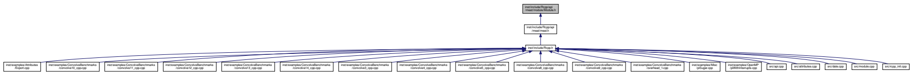
    Rcpp API.

Functions

- void Rcpp::message (SEXP s)

7.79 inst/include/Rcpp/api/meat/module/Module.h File Reference

This graph shows which files directly or indirectly include this file:



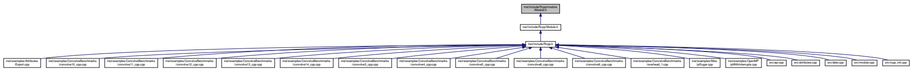
Namespaces

- Rcpp

    Rcpp API.

7.80 inst/include/Rcpp/module/Module.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/module/Module.h
inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::Module

Namespaces

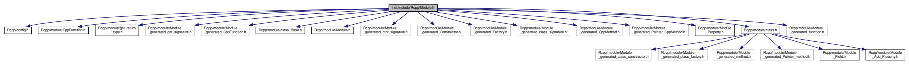
- Rcpp

  Rcpp API.

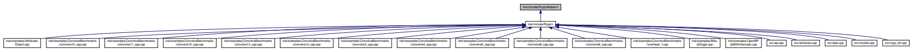
7.81 inst/include/Rcpp/Module.h File Reference

```
#include <Rcpp/config.h>
#include <Rcpp/module/CppFunction.h>
#include <Rcpp/module/get_return_type.h>
#include <Rcpp/module/Module_generated_get_signature.h>
#include <Rcpp/module/Module_generated_CppFunction.h>
#include <Rcpp/module/class_Base.h>
#include <Rcpp/module/Module.h>
#include <Rcpp/module/Module_generated_ctor_signature.h>
#include <Rcpp/module/Module_generated_Constructor.h>
#include <Rcpp/module/Module_generated_Factory.h>
#include <Rcpp/module/Module_generated_class_signature.h>
#include <Rcpp/module/Module_generated_CppMethod.h>
#include <Rcpp/module/Module_generated_Pointer_CppMethod.h>
#include <Rcpp/module/Module_Property.h>
#include <Rcpp/module/class.h>
#include <Rcpp/module/Module_generated_function.h>
```

Include dependency graph for Module.h:


inst/include/Rcpp/Module.h
Rcpp/config.h
Rcpp/module/CppFunction.h
Rcpp/module/get_return_type.h
Rcpp/module/Module_generated_get_signature.h
Rcpp/module/Module_generated_CppFunction.h
Rcpp/module/class_Base.h
Rcpp/module/Module.h
Rcpp/module/Module_generated_ctor_signature.h
Rcpp/module/Module_generated_Constructor.h
Rcpp/module/Module_generated_Factory.h
Rcpp/module/Module_generated_class_signature.h
Rcpp/module/Module_generated_CppMethod.h
Rcpp/module/Module_generated_Pointer_CppMethod.h
Rcpp/module/Module_Property.h
Rcpp/module/class.h
Rcpp/module/Module_generated_function.h
Rcpp/module/Module_generated_class_constructor.h
Rcpp/module/Module_generated_class_factory.h
Rcpp/module/Module_generated_method.h
Rcpp/module/Module_generated_Pointer_method.h
Rcpp/module/Module_Field.h
Rcpp/module/Module_Add_Property.h

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

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

7.81.1 Macro Definition Documentation

7.81.1.1 GET_MODULE_SYM

```
#define GET_MODULE_SYM ( moduleSym == NULL ? moduleSym = Rf_install("Module") :  moduleSym )
```

Definition at line 452 of file Module.h.

7.81.1.2 LOAD_RCPP_MODULE

```
#define LOAD_RCPP_MODULE(
                NAME )
```

Value:

```
    Shield<SEXP> __load_module_call__( Rf_lang2( GET_MODULE_SYM, _rcpp_module_boot_##NAME() ) ); \
    Rcpp_fast_eval( __load_module_call__, R_GlobalEnv );
```

Definition at line 455 of file Module.h.

7.81.1.3 RCPP_MODULE

```
#define RCPP_MODULE(
                name )
```

Value:

```
    void _rcpp_module_##name##_init() ;                                    \
    static Rcpp::Module _rcpp_module_##name( # name ) ;                    \
    extern "C" SEXP _rcpp_module_boot_##name(){                            \
        ::setCurrentScope( & _rcpp_module_##name ) ;                       \
        _rcpp_module_##name##_init( ) ;                                    \
        Rcpp::XPtr<Rcpp::Module> mod_xp(& _rcpp_module_##name , false);    \
        ::setCurrentScope( 0 ) ;                                           \
        return mod_xp ;                                                    \
    }                                                                      \
    void _rcpp_module_##name##_init()
```

Definition at line 428 of file Module.h.

7.81.1.4 RCPP_MODULE_BOOT

```
#define RCPP_MODULE_BOOT(
              name ) _rcpp_module_boot_##name
```

Definition at line 426 of file Module.h.

7.81.1.5 VARIABLE_IS_NOT_USED

```
#define VARIABLE_IS_NOT_USED
```

Definition at line 444 of file Module.h.

7.81.2 Variable Documentation

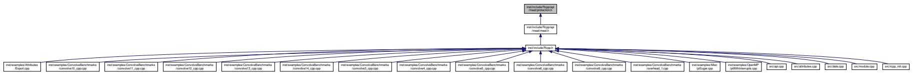
7.81.2.1 moduleSym

```
VARIABLE_IS_NOT_USED SEXP moduleSym = NULL  [static]
```

Definition at line 448 of file Module.h.

7.82 inst/include/Rcpp/api/meat/protection.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/protection.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

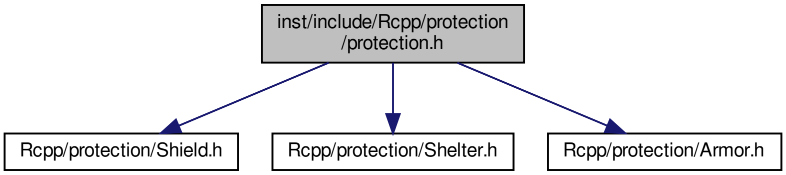
- Rcpp

  Rcpp API.

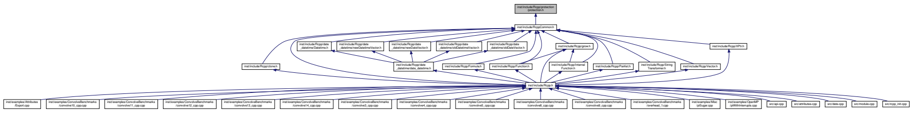
7.83 inst/include/Rcpp/protection/protection.h File Reference

```
#include <Rcpp/protection/Shield.h>
#include <Rcpp/protection/Shelter.h>
#include <Rcpp/protection/Armor.h>
```

Include dependency graph for protection.h:

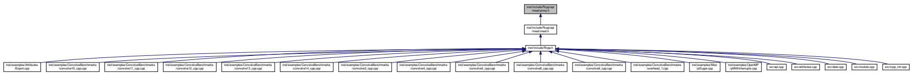


This graph shows which files directly or indirectly include this file:



7.84 inst/include/Rcpp/api/meat/proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

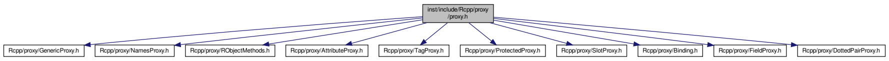
- Rcpp

  Rcpp API.

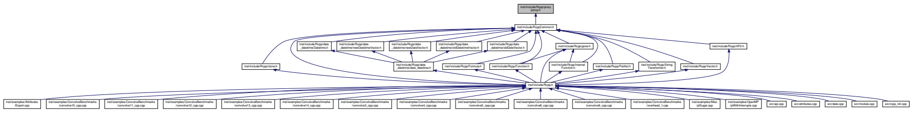
7.85 inst/include/Rcpp/proxy/proxy.h File Reference

```
#include <Rcpp/proxy/GenericProxy.h>
#include <Rcpp/proxy/NamesProxy.h>
#include <Rcpp/proxy/RObjectMethods.h>
#include <Rcpp/proxy/AttributeProxy.h>
#include <Rcpp/proxy/TagProxy.h>
#include <Rcpp/proxy/ProtectedProxy.h>
#include <Rcpp/proxy/SlotProxy.h>
#include <Rcpp/proxy/Binding.h>
#include <Rcpp/proxy/FieldProxy.h>
#include <Rcpp/proxy/DottedPairProxy.h>
```

Include dependency graph for proxy.h:

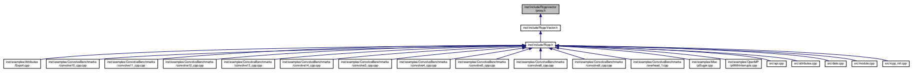


This graph shows which files directly or indirectly include this file:



7.86 inst/include/Rcpp/vector/proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >
- class Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >
- class Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_name_proxy< EXPRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy >

- struct Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_const_proxy< EXPRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy >
- struct Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >
- struct Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

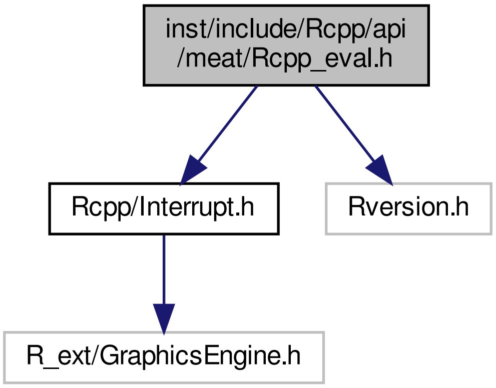
  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

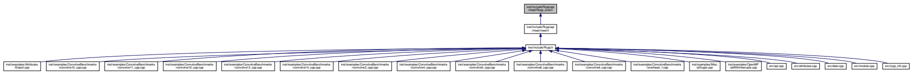
7.87 inst/include/Rcpp/api/meat/Rcpp_eval.h File Reference

```
#include <Rcpp/Interrupt.h>
#include <Rversion.h>
```

Include dependency graph for Rcpp_eval.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

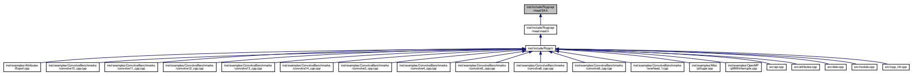
  internal implementation details

Functions

- SEXP Rcpp::internal::Rcpp_eval_impl (SEXP expr, SEXP env)
- SEXP Rcpp::Rcpp_fast_eval (SEXP expr, SEXP env)
- SEXP Rcpp::Rcpp_eval (SEXP expr, SEXP env)

7.88 inst/include/Rcpp/api/meat/S4.h File Reference

This graph shows which files directly or indirectly include this file:



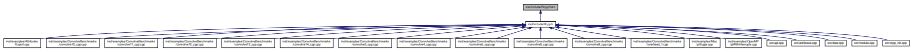
Namespaces

- Rcpp

  Rcpp API.

7.89 inst/include/Rcpp/S4.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

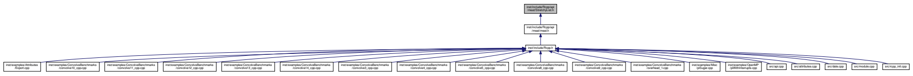
- typedef S4_Impl< PreserveStorage > Rcpp::S4

Functions

- Rcpp::RCPP_API_CLASS (S4_Impl)

7.90 inst/include/Rcpp/api/meat/StretchyList.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/StretchyList.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

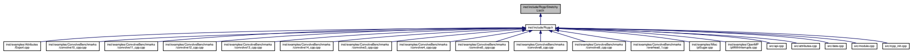
Namespaces

- Rcpp

  Rcpp API.

7.91 inst/include/Rcpp/StretchyList.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/StretchyList.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef StretchyList_Impl< PreserveStorage > Rcpp::StretchyList

Functions

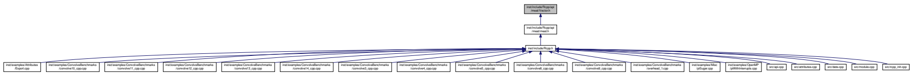
- Rcpp::RCPP_API_CLASS (StretchyList_Impl)
- Rcpp::StretchyList_Impl ()
- Rcpp::StretchyList_Impl (SEXP x)
- void Rcpp::update (SEXP)
- Rcpp::operator SEXP () const
- template<typename T >
  StretchyList_Impl & Rcpp::push_back (const T &obj)
- template<typename T >
  StretchyList_Impl & Rcpp::push_front (const T &obj)
- template<typename T >
  StretchyList_Impl & Rcpp::push_back__impl (const T &obj, traits::true_type)
- template<typename T >
  StretchyList_Impl & Rcpp::push_back__impl (const T &obj, traits::false_type)
- template<typename T >
  StretchyList_Impl & Rcpp::push_front__impl (const T &obj, traits::true_type)
- template<typename T >
  StretchyList_Impl & Rcpp::push_front__impl (const T &obj, traits::false_type)

Variables

- public Rcpp::DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > >

7.92 inst/include/Rcpp/api/meat/Vector.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/api/meat/Vector.h
inst/include/Rcpp/api/meat/meat.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

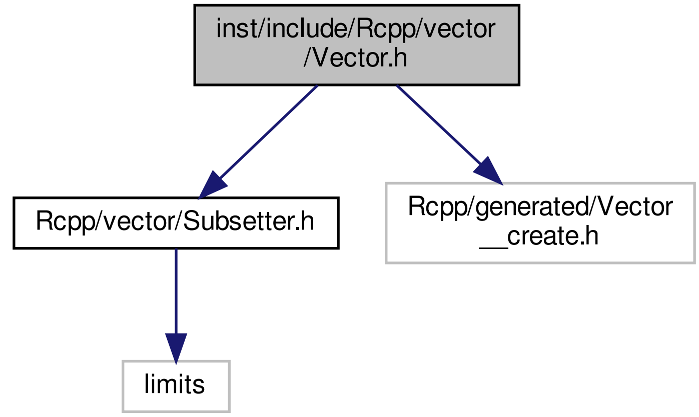
Functions

- template<typename T >
  SEXP Rcpp::internal::wrap_range_sugar_expression (const T &object, Rcpp::traits::true_type)

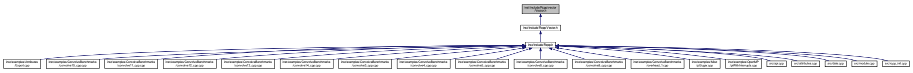
7.93 inst/include/Rcpp/vector/Vector.h File Reference

```
#include <Rcpp/vector/Subsetter.h>
#include <Rcpp/generated/Vector__create.h>
```

Include dependency graph for Vector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Vector< RTYPE, StoragePolicy >

Namespaces

- Rcpp

  Rcpp API.

Functions

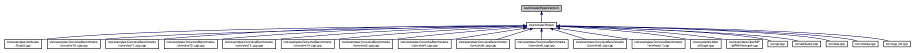
- template< int RTYPE, template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< RTYPE, StoragePolicy > &rhs)
- template< template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< STRSXP, StoragePolicy > &rhs)

7.94 inst/include/Rcpp/Vector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/exceptions.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/vector/00_forward_Vector.h>
#include <Rcpp/vector/no_init.h>
#include <Rcpp/vector/00_forward_proxy.h>
#include <Rcpp/vector/vector_from_string.h>
#include <Rcpp/vector/converter.h>
#include <Rcpp/vector/RangeIndexer.h>
#include <Rcpp/vector/Vector.h>
#include <Rcpp/vector/proxy.h>
#include <Rcpp/vector/traits.h>
#include <Rcpp/vector/DimNameProxy.h>
#include <Rcpp/vector/Matrix.h>
#include <Rcpp/vector/SubMatrix.h>
#include <Rcpp/vector/MatrixRow.h>
#include <Rcpp/vector/MatrixColumn.h>
#include <Rcpp/vector/instantiation.h>
#include <Rcpp/vector/string_proxy.h>
#include <Rcpp/vector/const_string_proxy.h>
#include <Rcpp/vector/generic_proxy.h>
#include <Rcpp/vector/const_generic_proxy.h>
#include <Rcpp/String.h>
#include <Rcpp/vector/LazyVector.h>
#include <Rcpp/vector/swap.h>
#include <Rcpp/vector/ChildVector.h>
#include <Rcpp/vector/ListOf.h>
```

Include dependency graph for Vector.h:

This graph shows which files directly or indirectly include this file:



Namespaces

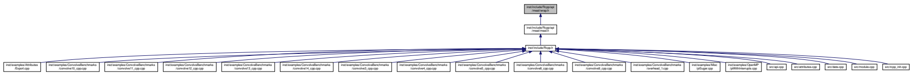
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.95 inst/include/Rcpp/api/meat/wrap.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

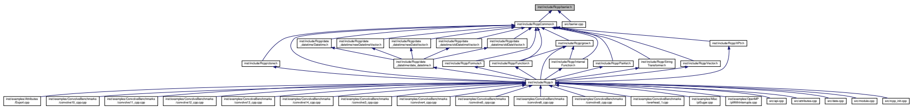
  internal implementation details

Functions

- template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
  SEXP Rcpp::internal::range_wrap_dispatch___impl__pair (InputIterator first, InputIterator last, Rcpp::traits::true_type)
- template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
  SEXP Rcpp::internal::range_wrap_dispatch___impl__pair (InputIterator first, InputIterator last, Rcpp::traits::false_type)

7.96 inst/include/Rcpp/barrier.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- SEXP get_string_elt (SEXP, R_xlen_t)
- const char ∗ char_get_string_elt (SEXP, R_xlen_t)
- void set_string_elt (SEXP, R_xlen_t, SEXP)
- void char_set_string_elt (SEXP, R_xlen_t, const char ∗)
- SEXP ∗ get_string_ptr (SEXP)
- SEXP get_vector_elt (SEXP, R_xlen_t)
- void set_vector_elt (SEXP, R_xlen_t, SEXP)
- SEXP ∗ get_vector_ptr (SEXP)
- const char ∗ char_nocheck (SEXP)
- void ∗ dataptr (SEXP)

7.96.1 Function Documentation

7.96.1.1 char_get_string_elt()

```
const char* char_get_string_elt (
             SEXP s,
             R_xlen_t i )  [inline]
```

Definition at line 216 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.2 char_nocheck()

```
const char* char_nocheck (
             SEXP x )  [inline]
```

Definition at line 258 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::String::setBuffer().

7.96.1.3 char_set_string_elt()

```
void char_set_string_elt (
            SEXP s,
            R_xlen_t i,
            const char * v )  [inline]
```

Definition at line 228 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.4 dataptr()

```
void* dataptr (
            SEXP x )  [inline]
```

Definition at line 264 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.5 get_string_elt()

```
SEXP get_string_elt (
            SEXP s,
            R_xlen_t i )  [inline]
```

Definition at line 210 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.6 get_string_ptr()

```
SEXP* get_string_ptr (
            SEXP s )  [inline]
```

Definition at line 234 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::sugar::get_const_begin(), and registerFunctions().

7.96.1.7 get_vector_elt()

```
SEXP get_vector_elt (
             SEXP v,
             R_xlen_t i )  [inline]
```

Definition at line 240 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.8 get_vector_ptr()

```
SEXP* get_vector_ptr (
             SEXP v )  [inline]
```

Definition at line 252 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.9 set_string_elt()

```
void set_string_elt (
             SEXP s,
             R_xlen_t i,
             SEXP v )  [inline]
```

Definition at line 222 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.96.1.10 set_vector_elt()

```
void set_vector_elt (
             SEXP v,
             R_xlen_t i,
             SEXP x )  [inline]
```

Definition at line 246 of file routines.h.

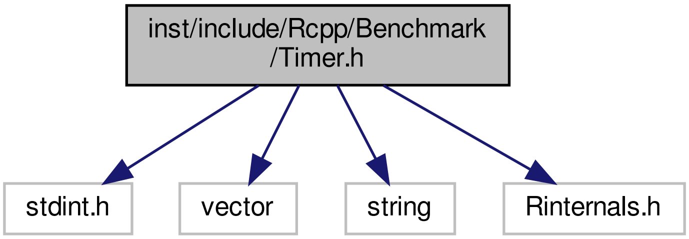
References GET_CALLABLE.

Referenced by registerFunctions().

7.97 inst/include/Rcpp/Benchmark/Timer.h File Reference

```
#include <stdint.h>
#include <vector>
#include <string>
#include <Rinternals.h>
```

Include dependency graph for Timer.h:



Classes

- class Rcpp::Timer

Namespaces

- Rcpp

    Rcpp API.

Macros

- #define R_NO_REMAP

Typedefs

- typedef uint64_t Rcpp::nanotime_t

7.97.1 Macro Definition Documentation

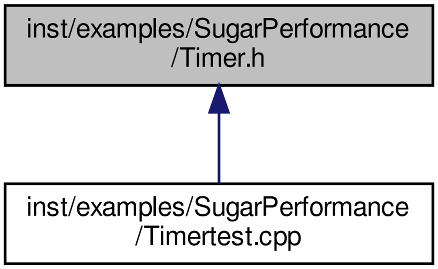
7.97.1.1 R_NO_REMAP

```
#define R_NO_REMAP
```

Definition at line 29 of file Timer.h.

7.98 inst/examples/SugarPerformance/Timer.h File Reference

This graph shows which files directly or indirectly include this file:


inst/examples/SugarPerformance
/Timer.h
inst/examples/SugarPerformance
/Timertest.cpp

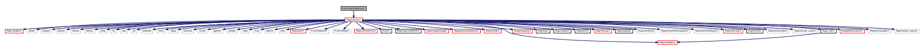
Classes

- class Timer

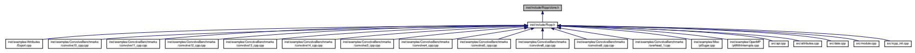
7.99 inst/include/Rcpp/clone.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for clone.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

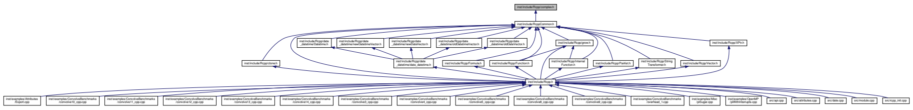
  Rcpp API.

Functions

- template<typename T >
  T Rcpp::clone (const T &object)

7.100 inst/include/Rcpp/complex.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define dplyr_tools_complex_H

Functions

- Rcomplex operator∗ (const Rcomplex &lhs, const Rcomplex &rhs)
- Rcomplex operator+ (const Rcomplex &lhs, const Rcomplex &rhs)
- Rcomplex operator- (const Rcomplex &lhs, const Rcomplex &rhs)
- Rcomplex operator/ (const Rcomplex &a, const Rcomplex &b)
- bool operator== (const Rcomplex &a, const Rcomplex &b)
- std::ostream & operator<< (std::ostream &os, const Rcomplex &cplx)

7.100.1 Macro Definition Documentation

7.100.1.1 dplyr_tools_complex_H

```
#define dplyr_tools_complex_H
```

Definition at line 74 of file complex.h.

7.100.2 Function Documentation

7.100.2.1 operator∗()

```
Rcomplex operator* (
            const Rcomplex & lhs,
            const Rcomplex & rhs )  [inline]
```

Definition at line 25 of file complex.h.

7.100.2.2 operator+()

```
Rcomplex operator+ (
            const Rcomplex & lhs,
            const Rcomplex & rhs )  [inline]
```

Definition at line 32 of file complex.h.

7.100.2.3 operator-()

```
Rcomplex operator- (
            const Rcomplex & lhs,
            const Rcomplex & rhs )  [inline]
```

Definition at line 39 of file complex.h.

7.100.2.4 operator/()

```
Rcomplex operator/ (
            const Rcomplex & a,
            const Rcomplex & b )  [inline]
```

Definition at line 46 of file complex.h.

7.100.2.5 operator<<()

```
std::ostream& operator<< (
             std::ostream & os,
             const Rcomplex & cplx )  [inline]
```

Definition at line 76 of file complex.h.

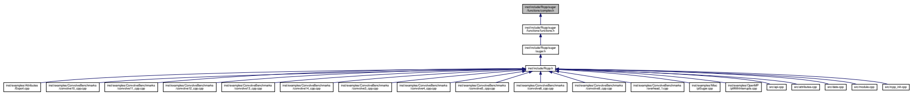
7.100.2.6 operator==()

```
bool operator== (
             const Rcomplex & a,
             const Rcomplex & b )  [inline]
```

Definition at line 68 of file complex.h.

7.101 inst/include/Rcpp/sugar/functions/complex.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/complex.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar
- Rcpp::internal

  internal implementation details

Macros

- #define RCPP_SUGAR_COMPLEX(__NAME__, __OUT__)

Functions

- double Rcpp::internal::complex__Re (Rcomplex x)
- double Rcpp::internal::complex__Im (Rcomplex x)
- double Rcpp::internal::complex__Mod (Rcomplex x)
- Rcomplex Rcpp::internal::complex__Conj (Rcomplex x)
- double Rcpp::internal::complex__Arg (Rcomplex x)
- Rcomplex Rcpp::internal::complex__exp (Rcomplex x)
- Rcomplex Rcpp::internal::complex__log (Rcomplex x)
- Rcomplex Rcpp::internal::complex__sqrt (Rcomplex z)
- Rcomplex Rcpp::internal::complex__cos (Rcomplex z)
- Rcomplex Rcpp::internal::complex__cosh (Rcomplex z)
- Rcomplex Rcpp::internal::complex__sin (Rcomplex z)
- Rcomplex Rcpp::internal::complex__tan (Rcomplex z)
- Rcomplex Rcpp::internal::complex__asin (Rcomplex z)
- Rcomplex Rcpp::internal::complex__acos (Rcomplex z)
- Rcomplex Rcpp::internal::complex__atan (Rcomplex z)
- Rcomplex Rcpp::internal::complex__acosh (Rcomplex z)
- Rcomplex Rcpp::internal::complex__asinh (Rcomplex z)
- Rcomplex Rcpp::internal::complex__atanh (Rcomplex z)
- Rcomplex Rcpp::internal::complex__sinh (Rcomplex z)
- Rcomplex Rcpp::internal::complex__tanh (Rcomplex z)

7.101.1 Macro Definition Documentation

7.101.1.1 RCPP_SUGAR_COMPLEX

```
#define RCPP_SUGAR_COMPLEX(
               __NAME__,
               __OUT__ )
```

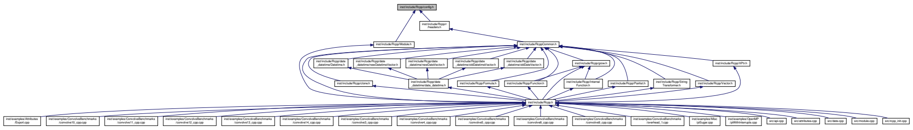
Value:

```
    template <bool NA, typename T>                                          \
    inline sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*)(Rcomplex) >        \
    __NAME__(const VectorBase<CPLXSXP,NA,T>& t) {                           \
        return sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*)(Rcomplex) >(  \
                                internal::complex__##__NAME__, t);          \
    }
```

Definition at line 231 of file complex.h.

7.102 inst/include/Rcpp/config.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/config.h
inst/include/Rcpp/r/headers.h
inst/include/Rcpp/Module.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define Rcpp_Version(v, p, s) (((v) ∗ 65536) + ((p) ∗ 256) + (s))
- #define RcppDevVersion(maj, min, rev, dev) (((maj)∗1000000) + ((min)∗10000) + ((rev)∗100) + (dev))
- #define RCPP_VERSION Rcpp_Version(1,0,9)
- #define RCPP_VERSION_STRING "1.0.9"
- #define RCPP_DEV_VERSION RcppDevVersion(1,0,9,0)
- #define RCPP_DEV_VERSION_STRING "1.0.9.0"

7.102.1 Macro Definition Documentation

7.102.1.1 RCPP_DEV_VERSION

```
#define RCPP_DEV_VERSION RcppDevVersion(1,0,9,0)
```

Definition at line 33 of file config.h.

7.102.1.2 RCPP_DEV_VERSION_STRING

```
#define RCPP_DEV_VERSION_STRING "1.0.9.0"
```

Definition at line 34 of file config.h.

7.102.1.3 Rcpp_Version

```
#define Rcpp_Version(
              v,
              p,
              s ) (((v) * 65536) + ((p) * 256) + (s))
```

Definition at line 24 of file config.h.

7.102.1.4 RCPP_VERSION

```
#define RCPP_VERSION Rcpp_Version(1,0,9)
```

Definition at line 29 of file config.h.

7.102.1.5 RCPP_VERSION_STRING

```
#define RCPP_VERSION_STRING "1.0.9"
```

Definition at line 30 of file config.h.

7.102.1.6 RcppDevVersion

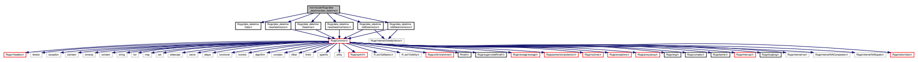
```
#define RcppDevVersion(
             maj,
             min,
             rev,
             dev ) (((maj)*1000000) + ((min)*10000) + ((rev)*100) + (dev))
```

Definition at line 26 of file config.h.

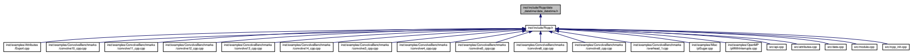
7.103 inst/include/Rcpp/date_datetime/date_datetime.h File Reference

```
#include <Rcpp/date_datetime/Date.h>
#include <Rcpp/date_datetime/oldDateVector.h>
#include <Rcpp/date_datetime/newDateVector.h>
#include <Rcpp/date_datetime/Datetime.h>
#include <Rcpp/date_datetime/oldDatetimeVector.h>
#include <Rcpp/date_datetime/newDatetimeVector.h>
```

Include dependency graph for date_datetime.h:



This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.

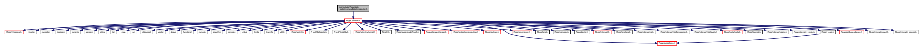
Typedefs

- typedef oldDateVector Rcpp::DateVector
- typedef oldDatetimeVector Rcpp::DatetimeVector

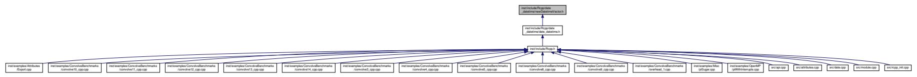
7.104 inst/include/Rcpp/date_datetime/newDatetimeVector.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for newDatetimeVector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::newDatetimeVector

Namespaces

- Rcpp

  Rcpp API.

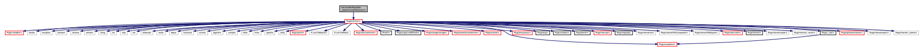
Functions

- std::ostream & Rcpp::operator<< (std::ostream &os, const newDatetimeVector d)

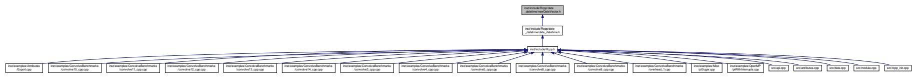
7.105 inst/include/Rcpp/date_datetime/newDateVector.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for newDateVector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::newDateVector

Namespaces

- Rcpp

  Rcpp API.

Functions

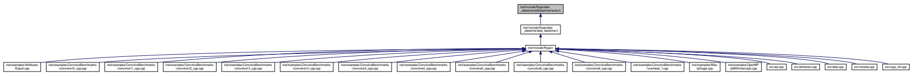
- std::ostream & Rcpp::operator<< (std::ostream &os, const newDateVector d)

7.106 inst/include/Rcpp/date_datetime/oldDatetimeVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
```

Include dependency graph for oldDatetimeVector.h:

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::oldDatetimeVector

Namespaces

- Rcpp

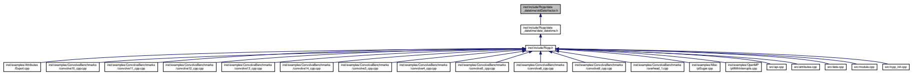
  Rcpp API.

7.107 inst/include/Rcpp/date_datetime/oldDateVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
```

Include dependency graph for oldDateVector.h:

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::oldDateVector

Namespaces

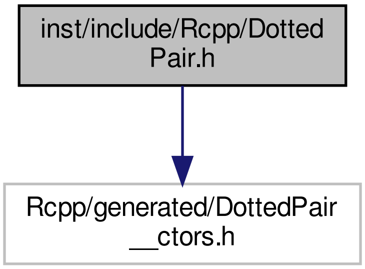
- Rcpp

    Rcpp API.

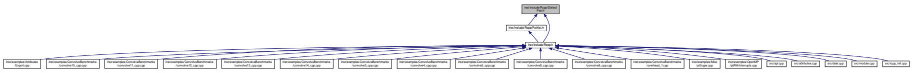
7.108 inst/include/Rcpp/DottedPair.h File Reference

```
#include <Rcpp/generated/DottedPair__ctors.h>
```

Include dependency graph for DottedPair.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef DottedPair_Impl< PreserveStorage > Rcpp::DottedPair

Functions

- Rcpp::RCPP_API_CLASS (DottedPair_Impl)

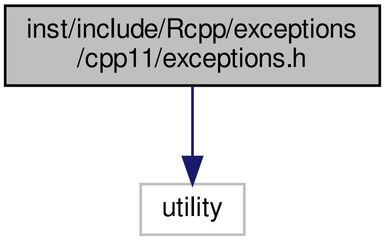
Variables

- public Rcpp::DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< DottedPair_Impl< StoragePolicy > >

7.109 inst/include/Rcpp/exceptions/cpp11/exceptions.h File Reference

```
#include <utility>
```

Include dependency graph for exceptions.h:



Namespaces

- Rcpp

  Rcpp API.

Macros

- #define RCPP_ADVANCED_EXCEPTION_CLASS(__CLASS__, __WHAT__)

Functions

- template<typename... Args>
  void Rcpp::warning (const char ∗fmt, Args &&... args)
- template<typename... Args>
  void NORET Rcpp::stop (const char ∗fmt, Args &&... args)

7.109.1 Macro Definition Documentation

7.109.1.1 RCPP_ADVANCED_EXCEPTION_CLASS

```
#define RCPP_ADVANCED_EXCEPTION_CLASS(
                __CLASS__,
                __WHAT__ )
```

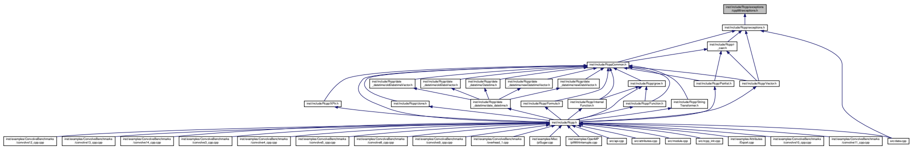
Value:

```
class __CLASS__ : public std::exception {                                        \
    public:                                                                      \
        __CLASS__( ) throw() : message( std::string(__WHAT__) + "." ){}          \
        __CLASS__( const std::string& message ) throw() :                        \
                message( std::string(__WHAT__) + ": " + message + "."){}                \
        template <typename... Args>                                              \
        __CLASS__( const char* fmt, Args&&... args ) throw() :                   \
            message(  tfm::format(fmt, std::forward<Args>(args)... ) ){}         \
        virtual ~__CLASS__() throw(){}                                           \
        virtual const char* what() const throw() { return message.c_str(); }     \
        private:                                                                 \
            std::string message;                                                 \
};
```

Definition at line 30 of file exceptions.h.

7.110 inst/include/Rcpp/exceptions/cpp98/exceptions.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.

Macros

- #define RCPP_ADVANCED_EXCEPTION_CLASS(__CLASS__, __WHAT__)

Functions

- template<typename T1 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1)
- template<typename T1 , typename T2 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1 , typename T2 , typename T3 , typename T4 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >
  void Rcpp::warning (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- template<typename T1 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1)
- template<typename T1 , typename T2 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1 , typename T2 , typename T3 , typename T4 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)

- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >
  void NORET Rcpp::stop (const char ∗fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)

7.110.1 Macro Definition Documentation

7.110.1.1 RCPP_ADVANCED_EXCEPTION_CLASS

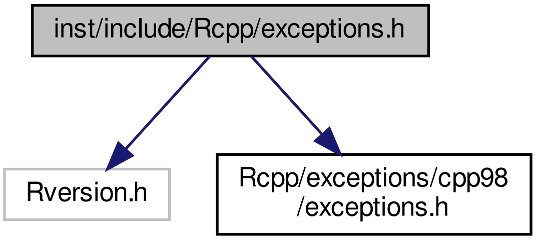
```
#define RCPP_ADVANCED_EXCEPTION_CLASS(
            __CLASS__,
            __WHAT__ )
```

Definition at line 28 of file exceptions.h.

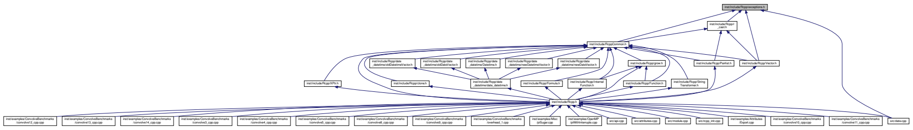
7.111 inst/include/Rcpp/exceptions.h File Reference

```
#include <Rversion.h>
#include <Rcpp/exceptions/cpp98/exceptions.h>
```

Include dependency graph for exceptions.h:



This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/exceptions.h
inst/include/Rcpp/r_cast.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp.h
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/module.cpp
src/rcpp_init.cpp
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
src/date.cpp

Classes

- class Rcpp::exception
- class Rcpp::no_such_env
- class Rcpp::file_io_error
- class Rcpp::file_not_found
- class Rcpp::file_exists
- struct Rcpp::LongjumpException

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Macros

- #define RCPP_DEFAULT_INCLUDE_CALL true
- #define GET_STACKTRACE() R_NilValue
- #define RCPP_EXCEPTION_CLASS(__CLASS__, __WHAT__)
- #define RCPP_SIMPLE_EXCEPTION_CLASS(__CLASS__, __MESSAGE__)
- #define DEMANGLE(__TYPE__) demangle(typeid(__TYPE__).name()).c_str()

Functions

- static std::string Rcpp::toString (const int i)
- void Rcpp::warning (const std::string &message)
- void NORET Rcpp::stop (const std::string &message)
- SEXP Rcpp::internal::longjumpSentinel (SEXP token)
- bool Rcpp::internal::isLongjumpSentinel (SEXP x)
- SEXP Rcpp::internal::getLongjumpToken (SEXP sentinel)
- void Rcpp::internal::resumeJump (SEXP token)
- Rcpp::RCPP_SIMPLE_EXCEPTION_CLASS (not_a_matrix, "Not a matrix.") RCPP_SIMPLE_EXCEPTION_↩ CLASS(no_such_field
- No such field Rcpp::RCPP_EXCEPTION_CLASS (reference_creation_error, "Error creating object of reference class") RCPP_ADVANCED_EXCEPTION_CLASS(not_compatible
- SEXP Rcpp::internal::nth (SEXP s, int n)

- bool Rcpp::internal::is_Rcpp_eval_call (SEXP expr)
- SEXP get_last_call ()
- SEXP get_exception_classes (const std::string &ex_class)
- SEXP make_condition (const std::string &ex_msg, SEXP call, SEXP cppstack, SEXP classes)
- template<typename Exception >
  SEXP exception_to_condition_template (const Exception &ex, bool include_call)
- SEXP rcpp_exception_to_r_condition (const Rcpp::exception &ex)
- SEXP exception_to_r_condition (const std::exception &ex)
- SEXP string_to_try_error (const std::string &str)
- SEXP exception_to_try_error (const std::exception &ex)
- std::string demangle (const std::string &name)
- void forward_exception_to_r (const std::exception &ex)
- void forward_rcpp_exception_to_r (const Rcpp::exception &ex)

7.111.1 Macro Definition Documentation

7.111.1.1 DEMANGLE

```
#define DEMANGLE(
              __TYPE__ ) demangle( typeid(__TYPE__).name() ).c_str()
```

Definition at line 382 of file exceptions.h.

7.111.1.2 GET_STACKTRACE

```
#define GET_STACKTRACE( ) R_NilValue
```

Definition at line 31 of file exceptions.h.

7.111.1.3 RCPP_DEFAULT_INCLUDE_CALL

```
#define RCPP_DEFAULT_INCLUDE_CALL true
```

Definition at line 28 of file exceptions.h.

7.111.1.4 RCPP_EXCEPTION_CLASS

```
#define RCPP_EXCEPTION_CLASS(
              __CLASS__,
              __WHAT__ )
```

Value:

```
    class __CLASS__ : public std::exception{                                       \
    public:                                                                        \
        __CLASS__( ) throw() : message( std::string(__WHAT__) + "." ){} ;          \
        __CLASS__( const std::string& message ) throw() :                          \
        message( std::string(__WHAT__) + ": " + message + "." ){} ;                \
        virtual ~__CLASS__() throw(){} ;                                           \
        virtual const char* what() const throw() { return message.c_str() ; }      \
    private:                                                                       \
        std::string message ;                                                      \
    } ;
```

Definition at line 187 of file exceptions.h.

7.111.1.5 RCPP_SIMPLE_EXCEPTION_CLASS

```
#define RCPP_SIMPLE_EXCEPTION_CLASS(
              __CLASS__,
              __MESSAGE__ )
```

Value:

```
    class __CLASS__ : public std::exception{                                       \
    public:                                                                        \
        __CLASS__() throw() {} ;                                                   \
        virtual ~__CLASS__() throw(){} ;                                           \
        virtual const char* what() const throw() { return __MESSAGE__ ; }          \
    } ;
```

Definition at line 199 of file exceptions.h.

7.111.2 Function Documentation

7.111.2.1 demangle()

```
std::string demangle (
             const std::string & name )   [inline]
```

Definition at line 192 of file routines.h.

References GET_CALLABLE.

Referenced by exception_to_condition_template(), Rcpp::get_return_type_dispatch(), and registerFunctions().

7.111.2.2 exception_to_condition_template()

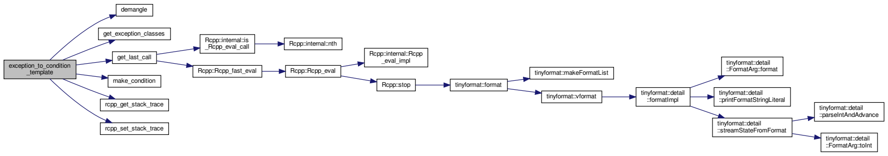
```
template<typename Exception >
SEXP exception_to_condition_template (
            const Exception & ex,
            bool include_call )  [inline]
```

Definition at line 324 of file exceptions.h.

References demangle(), get_exception_classes(), get_last_call(), make_condition(), rcpp_get_stack_trace(), and rcpp_set_stack_trace().

Referenced by exception_to_r_condition(), and rcpp_exception_to_r_condition().

Here is the call graph for this function:


exception_to_condition _template
demangle
get_exception_classes
get_last_call
make_condition
rcpp_get_stack_trace
rcpp_set_stack_trace
Rcpp::internal::is _Rcpp_eval_call
Rcpp::internal::nth
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

7.111.2.3 exception_to_r_condition()

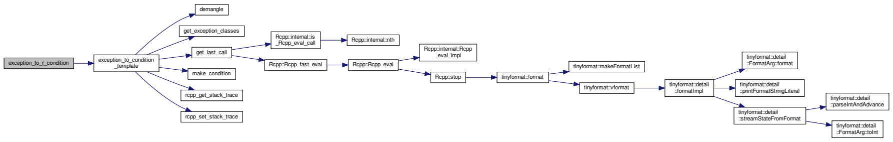
```
SEXP exception_to_r_condition (
            const std::exception & ex )  [inline]
```

Definition at line 352 of file exceptions.h.

References exception_to_condition_template(), and RCPP_DEFAULT_INCLUDE_CALL.

Referenced by forward_exception_to_r(), and forward_rcpp_exception_to_r().

Here is the call graph for this function:


exception_to_r_condition
exception_to_condition _template
demangle
get_exception_classes
get_last_call
make_condition
rcpp_get_stack_trace
rcpp_set_stack_trace
Rcpp::internal::is _Rcpp_eval_call
Rcpp::internal::nth
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

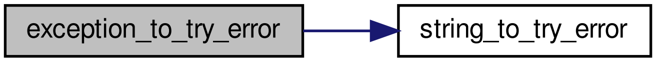
7.111.2.4 exception_to_try_error()

```
SEXP exception_to_try_error (
            const std::exception & ex )  [inline]
```

Definition at line 376 of file exceptions.h.

References string_to_try_error().

Here is the call graph for this function:



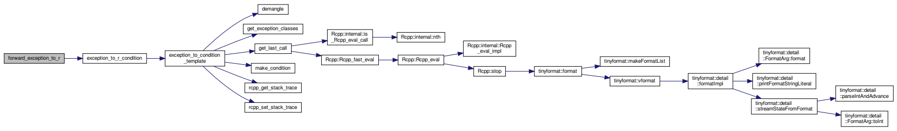
7.111.2.5 forward_exception_to_r()

```
void forward_exception_to_r (
            const std::exception & ex )  [inline]
```

Definition at line 386 of file exceptions.h.

References exception_to_r_condition().

Here is the call graph for this function:


forward_exception_to_r
exception_to_r_condition
exception_to_condition_template
demangle
get_exception_classes
get_last_call
make_condition
rcpp_get_stack_trace
rcpp_set_stack_trace
Rcpp::internal::is_Rcpp_eval_call
Rcpp::internal::nth
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp_eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail::formatImpl
tinyformat::detail::FormatArg::format
tinyformat::detail::printFormatStringLiteral
tinyformat::detail::streamStateFromFormat
tinyformat::detail::parseIntAndAdvance
tinyformat::detail::FormatArg::toInt

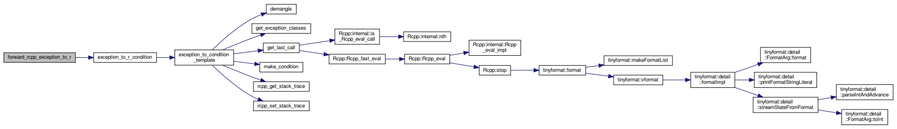
7.111.2.6 forward_rcpp_exception_to_r()

```
void forward_rcpp_exception_to_r (
            const Rcpp::exception & ex )  [inline]
```

Definition at line 393 of file exceptions.h.

References exception_to_r_condition().

Here is the call graph for this function:


forward_rcpp_exception_to_r
exception_to_r_condition
exception_to_condition _template
demangle
get_exception_classes
get_last_call
make_condition
rcpp_get_stack_trace
rcpp_set_stack_trace
Rcpp::internal::is _Rcpp_eval_call
Rcpp::internal::nth
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

7.111.2.7 get_exception_classes()

```
SEXP get_exception_classes (
            const std::string & ex_class )  [inline]
```

Definition at line 288 of file exceptions.h.

Referenced by exception_to_condition_template().

7.111.2.8 get_last_call()

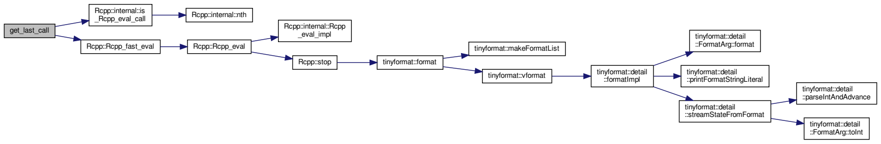
```
SEXP get_last_call ( )  [inline]
```

Definition at line 268 of file exceptions.h.

References Rcpp::internal::is_Rcpp_eval_call(), and Rcpp::Rcpp_fast_eval().

Referenced by exception_to_condition_template().

Here is the call graph for this function:


get_last_call
Rcpp::internal::is _Rcpp_eval_call
Rcpp::internal::nth
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
Rcpp::internal::Rcpp _eval_impl
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt

7.111.2.9 make_condition()

```
SEXP make_condition (
            const std::string & ex_msg,
            SEXP call,
            SEXP cppstack,
            SEXP classes )  [inline]
```

Definition at line 302 of file exceptions.h.

Referenced by exception_to_condition_template().

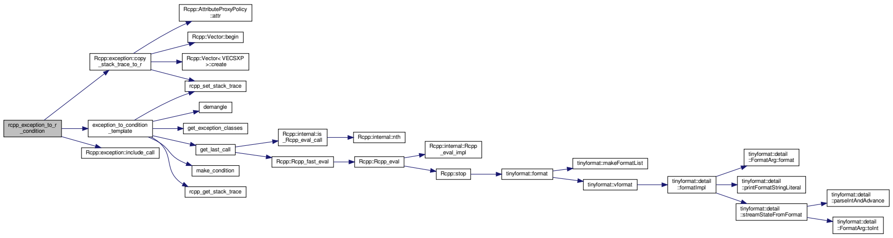
7.111.2.10 rcpp_exception_to_r_condition()

```
SEXP rcpp_exception_to_r_condition (
            const Rcpp::exception & ex )  [inline]
```

Definition at line 347 of file exceptions.h.

References Rcpp::exception::copy_stack_trace_to_r(), exception_to_condition_template(), and Rcpp::exception::include_call().

Here is the call graph for this function:


Rcpp::AttributeProxyPolicy ::attr
Rcpp::Vector::begin
Rcpp::exception::copy _stack_trace_to_r
Rcpp::Vector< VECSXP >::create
rcpp_set_stack_trace
demangle
rcpp_exception_to_r _condition
exception_to_condition _template
get_exception_classes
Rcpp::internal::is _Rcpp_eval_call
Rcpp::internal::nth
Rcpp::exception::include_call
get_last_call
Rcpp::internal::Rcpp _eval_impl
Rcpp::Rcpp_fast_eval
Rcpp::Rcpp_eval
make_condition
Rcpp::stop
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt
rcpp_get_stack_trace

7.111.2.11 string_to_try_error()

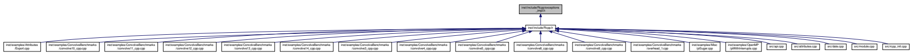
```
SEXP string_to_try_error (
            const std::string & str )  [inline]
```

Definition at line 356 of file exceptions.h.

Referenced by exception_to_try_error().

7.112 inst/include/Rcpp/exceptions_impl.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.

Macros

- #define RCPP_DEMANGLER_ENABLED 0

7.112.1 Macro Definition Documentation

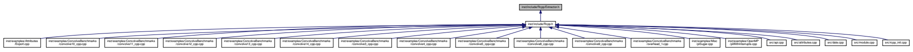
7.112.1.1 RCPP_DEMANGLER_ENABLED

```
#define RCPP_DEMANGLER_ENABLED 0
```

Definition at line 41 of file exceptions_impl.h.

7.113 inst/include/Rcpp/Extractor.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::Extractor< RTYPE, NA, VECTOR >

Namespaces

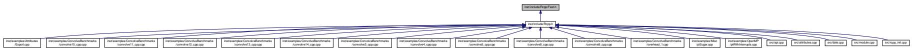
- Rcpp

    Rcpp API.
- Rcpp::traits

    traits used to dispatch wrap

7.114 inst/include/Rcpp/Fast.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Fast< VECTOR >

Namespaces

- Rcpp

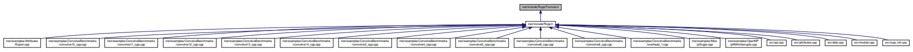
    Rcpp API.

7.115 inst/include/Rcpp/Formula.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/Language.h>
```

Include dependency graph for Formula.h:

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.

Typedefs

- typedef Formula_Impl< PreserveStorage > Rcpp::Formula

Functions

- Rcpp::RCPP_API_CLASS (Formula_Impl)
- Rcpp::Formula_Impl (SEXP x)
- Rcpp::Formula_Impl (const std::string &code)
- void Rcpp::update (SEXP)

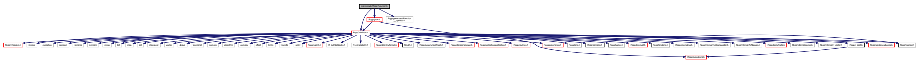
Variables

- public Rcpp::DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< Formula_Impl< StoragePolicy > >

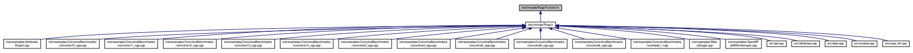
7.116 inst/include/Rcpp/Function.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/grow.h>
#include <Rcpp/generated/Function__operator.h>
```

Include dependency graph for Function.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef Function_Impl< PreserveStorage > Rcpp::Function

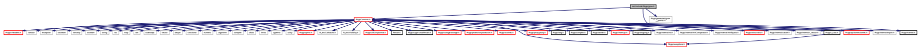
Functions

- Rcpp::RCPP_API_CLASS (Function_Impl)

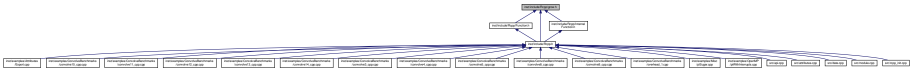
7.117 inst/include/Rcpp/grow.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/Named.h>
#include <Rcpp/generated/grow__pairlist.h>
```

Include dependency graph for grow.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

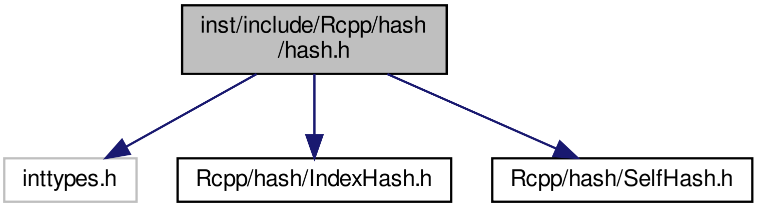
Functions

- SEXP Rcpp::pairlist ()
- SEXP Rcpp::grow (SEXP head, SEXP tail)
- template<typename T >
  SEXP Rcpp::internal::grow__dispatch (::Rcpp::traits::false_type, const T &head, SEXP tail)
- template<typename T >
  SEXP Rcpp::internal::grow__dispatch (::Rcpp::traits::true_type, const T &head, SEXP tail)
- template<typename T >
  SEXP Rcpp::grow (const T &head, SEXP tail)
- SEXP Rcpp::grow (const char ∗head, SEXP tail)

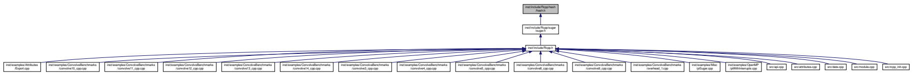
7.118 inst/include/Rcpp/hash/hash.h File Reference

```
#include <inttypes.h>
#include <Rcpp/hash/IndexHash.h>
#include <Rcpp/hash/SelfHash.h>
```

Include dependency graph for hash.h:

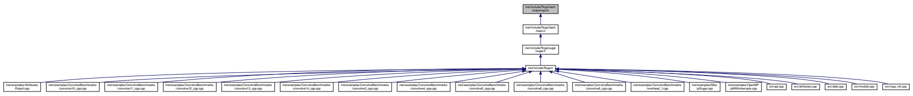


This graph shows which files directly or indirectly include this file:



7.119 inst/include/Rcpp/hash/IndexHash.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::IndexHash< RTYPE >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define RCPP_PROFILE_TIC
- #define RCPP_PROFILE_TOC
- #define RCPP_PROFILE_RECORD(name)
- #define RCPP_USE_CACHE_HASH
- #define RCPP_HASH(X) (3141592653U ∗ ((uint32_t)(X)) >> (32 - k))

7.119.1 Macro Definition Documentation

7.119.1.1 RCPP_HASH

```
#define RCPP_HASH(
             X ) (3141592653U * ((uint32_t)(X)) >> (32 - k))
```

Definition at line 45 of file IndexHash.h.

7.119.1.2 RCPP_PROFILE_RECORD

```
#define RCPP_PROFILE_RECORD(
             name )
```

Definition at line 37 of file IndexHash.h.

7.119.1.3 RCPP_PROFILE_TIC

```
#define RCPP_PROFILE_TIC
```

Definition at line 35 of file IndexHash.h.

7.119.1.4 RCPP_PROFILE_TOC

```
#define RCPP_PROFILE_TOC
```

Definition at line 36 of file IndexHash.h.

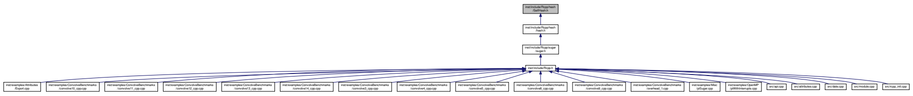
7.119.1.5 RCPP_USE_CACHE_HASH

```
#define RCPP_USE_CACHE_HASH
```

Definition at line 39 of file IndexHash.h.

7.120 inst/include/Rcpp/hash/SelfHash.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SelfHash< RTYPE >

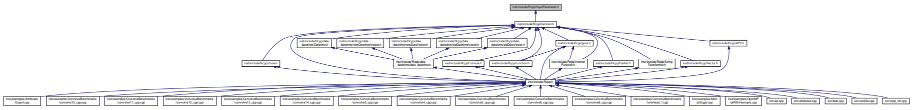
Namespaces

- Rcpp

    Rcpp API.
- Rcpp::sugar

7.121 inst/include/Rcpp/InputParameter.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/InputParameter.h
inst/include/RcppCommon.h
inst/include/Rcpp/date
_datetime/Datetime.h
inst/include/Rcpp/date
_datetime/newDatetimeVector.h
inst/include/Rcpp/date
_datetime/newDateVector.h
inst/include/Rcpp/date
_datetime/oldDatetimeVector.h
inst/include/Rcpp/date
_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date
_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal
Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String
Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes
/Export.cpp
inst/examples/ConvolveBenchmarks
/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks
/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks
/overhead_1.cpp
inst/examples/Misc
/piSugar.cpp
inst/examples/OpenMP
/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::InputParameter< T >
- class Rcpp::ReferenceInputParameter< T >
- class Rcpp::ConstInputParameter< T >
- class Rcpp::ConstReferenceInputParameter< T >
- struct Rcpp::traits::input_parameter< T >
- struct Rcpp::traits::input_parameter< T & >
- struct Rcpp::traits::input_parameter< const T >
- struct Rcpp::traits::input_parameter< const T & >

Namespaces

- Rcpp

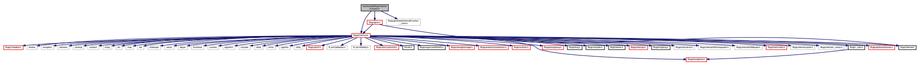
  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

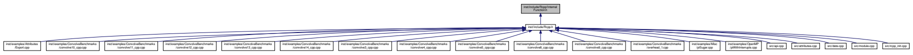
7.122 inst/include/Rcpp/InternalFunction.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/grow.h>
#include <Rcpp/generated/InternalFunction__ctors.h>
```

Include dependency graph for InternalFunction.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef InternalFunction_Impl< PreserveStorage > Rcpp::InternalFunction

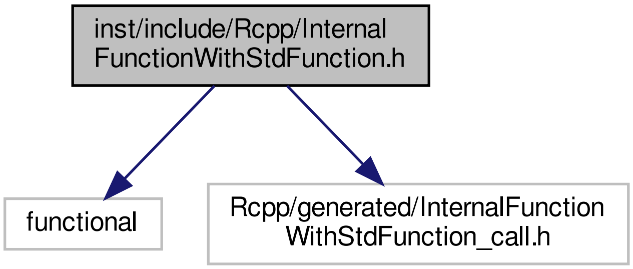
Functions

- Rcpp::RCPP_API_CLASS (InternalFunction_Impl)

7.123 inst/include/Rcpp/InternalFunctionWithStdFunction.h File Reference

```
#include <functional>
#include <Rcpp/generated/InternalFunctionWithStdFunction_call.h>
```

Include dependency graph for InternalFunctionWithStdFunction.h:


inst/include/Rcpp/Internal
FunctionWithStdFunction.h
functional
Rcpp/generated/InternalFunction
WithStdFunction_call.h

Classes

- class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >
- class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >

Namespaces

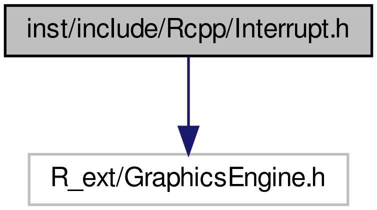
- Rcpp

  Rcpp API.
- Rcpp::InternalFunctionWithStdFunction

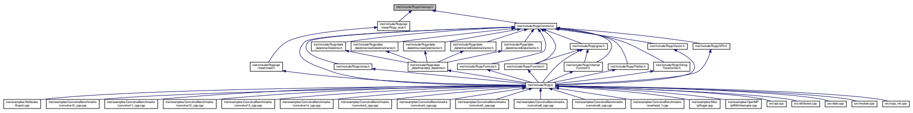
7.124 inst/include/Rcpp/Interrupt.h File Reference

```
#include <R_ext/GraphicsEngine.h>
```

Include dependency graph for Interrupt.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::InterruptedException

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

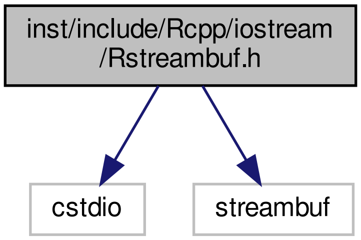
Functions

- SEXP Rcpp::internal::interruptedError ()
- void Rcpp::checkUserInterrupt ()

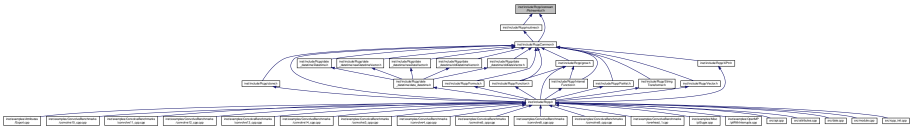
7.125 inst/include/Rcpp/iostream/Rstreambuf.h File Reference

```
#include <cstdio>
#include <streambuf>
```

Include dependency graph for Rstreambuf.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Rstreambuf< OUTPUT >
- class Rcpp::Rostream< OUTPUT >

Namespaces

- Rcpp

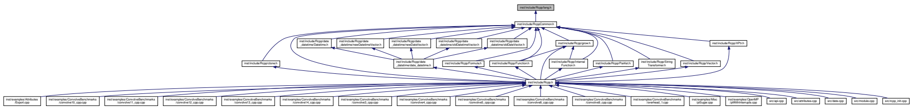
  Rcpp API.

Variables

- static Rostream< true > Rcpp::Rcout = Rcpp_cout_get()
- static Rostream< false > Rcpp::Rcerr = Rcpp_cerr_get()

7.126 inst/include/Rcpp/lang.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Macros

- #define Rcpp_list1 Rf_list1
- #define Rcpp_lang1 Rf_lang1
- #define Rcpp_lang2 Rf_lang2
- #define Rcpp_lang3 Rf_lang3
- #define Rcpp_lang4 Rf_lang4
- #define Rcpp_lang5 Rf_lang5
- #define Rcpp_lang6 Rf_lang6
- #define Rcpp_lcons Rf_lcons

Functions

- SEXP Rcpp::Rcpp_list2 (SEXP x0, SEXP x1)
- SEXP Rcpp::Rcpp_list3 (SEXP x0, SEXP x1, SEXP x2)
- SEXP Rcpp::Rcpp_list4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP Rcpp::Rcpp_list5 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4)
- SEXP Rcpp::Rcpp_list6 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5)
- SEXP Rcpp::Rcpp_list7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP Rcpp::Rcpp_lang7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP Rcpp::Rcpp_list8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP Rcpp::Rcpp_lang8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP Rcpp::Rcpp_list9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)
- SEXP Rcpp::Rcpp_lang9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)
- SEXP Rcpp::Rcpp_list10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)
- SEXP Rcpp::Rcpp_lang10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)
- SEXP Rcpp::Rcpp_list11 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10)

- SEXP Rcpp::Rcpp_lang11 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10)
- SEXP Rcpp::Rcpp_list12 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11)
- SEXP Rcpp::Rcpp_lang12 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11)
- SEXP Rcpp::Rcpp_list13 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12)
- SEXP Rcpp::Rcpp_lang13 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12)
- SEXP Rcpp::Rcpp_list14 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13)
- SEXP Rcpp::Rcpp_lang14 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13)
- SEXP Rcpp::Rcpp_list15 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14)
- SEXP Rcpp::Rcpp_lang15 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14)
- SEXP Rcpp::Rcpp_list16 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15)
- SEXP Rcpp::Rcpp_lang16 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15)
- SEXP Rcpp::Rcpp_list17 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16)
- SEXP Rcpp::Rcpp_lang17 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16)
- SEXP Rcpp::Rcpp_list18 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17)
- SEXP Rcpp::Rcpp_lang18 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17)
- SEXP Rcpp::Rcpp_list19 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18)
- SEXP Rcpp::Rcpp_lang19 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18)
- SEXP Rcpp::Rcpp_list20 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18, SEXP x19)
- SEXP Rcpp::Rcpp_lang20 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18, SEXP x19)

7.126.1 Macro Definition Documentation

7.126.1.1 Rcpp_lang1

```
#define Rcpp_lang1 Rf_lang1
```

Definition at line 25 of file lang.h.

7.126.1.2 Rcpp_lang2

```
#define Rcpp_lang2 Rf_lang2
```

Definition at line 26 of file lang.h.

7.126.1.3 Rcpp_lang3

```
#define Rcpp_lang3 Rf_lang3
```

Definition at line 27 of file lang.h.

7.126.1.4 Rcpp_lang4

```
#define Rcpp_lang4 Rf_lang4
```

Definition at line 28 of file lang.h.

7.126.1.5 Rcpp_lang5

```
#define Rcpp_lang5 Rf_lang5
```

Definition at line 29 of file lang.h.

7.126.1.6 Rcpp_lang6

```
#define Rcpp_lang6 Rf_lang6
```

Definition at line 30 of file lang.h.

7.126.1.7 Rcpp_lcons

```
#define Rcpp_lcons Rf_lcons
```

Definition at line 32 of file lang.h.

7.126.1.8 Rcpp_list1

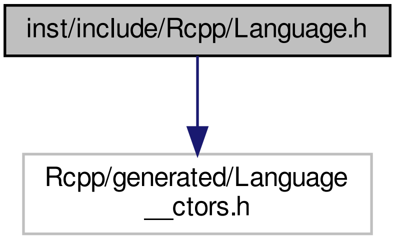
```
#define Rcpp_list1 Rf_list1
```

Definition at line 24 of file lang.h.

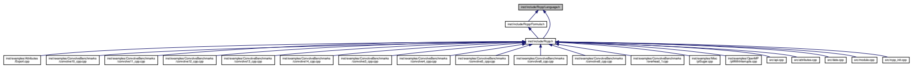
7.127 inst/include/Rcpp/Language.h File Reference

```
#include <Rcpp/generated/Language__ctors.h>
```

Include dependency graph for Language.h:


inst/include/Rcpp/Language.h
Rcpp/generated/Language__ctors.h

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Language.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::fixed_call< RESULT_TYPE >
- class Rcpp::unary_call< T, RESULT_TYPE >
- class Rcpp::binary_call< T1, T2, RESULT_TYPE >

Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef DottedPairProxyPolicy< Language_Impl >::const_DottedPairProxy Rcpp::const_Proxy
- typedef Language_Impl< PreserveStorage > Rcpp::Language

Functions

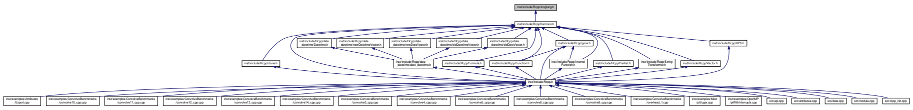
- Rcpp::RCPP_API_CLASS (Language_Impl)
- Rcpp::Language_Impl ()
- Rcpp::Language_Impl (SEXP x)
- Rcpp::Language_Impl (const std::string &symbol)
- Rcpp::Language_Impl (const Symbol &symbol)
- Rcpp::Language_Impl (const Function &function)
- void Rcpp::setSymbol (const std::string &symbol)
- void Rcpp::setSymbol (const Symbol &symbol)
- void Rcpp::setFunction (const Function &function)
- SEXP Rcpp::eval () const
- SEXP Rcpp::eval (SEXP env) const
- SEXP Rcpp::fast_eval () const
- SEXP Rcpp::fast_eval (SEXP env) const
- void Rcpp::update (SEXP)

Variables

- public Rcpp::DottedPairProxyPolicy< Language_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< Language_Impl< StoragePolicy > >

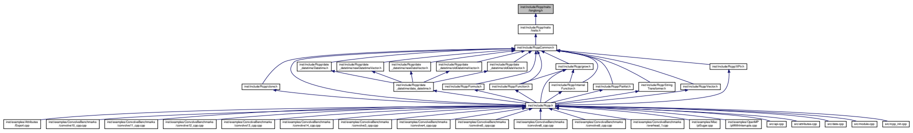
7.128 inst/include/Rcpp/longlong.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/longlong.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

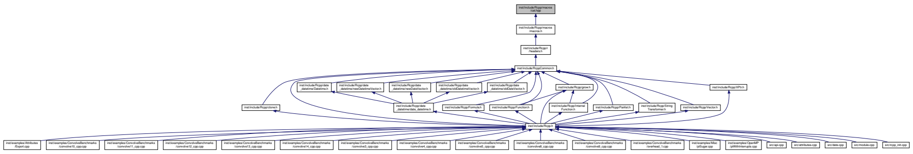
7.129 inst/include/Rcpp/traits/longlong.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/longlong.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

7.130 inst/include/Rcpp/macros/cat.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_PP_CAT(a, b) RCPP_PP_CAT_OO((a, b))
- #define RCPP_PP_CAT_OO(par) RCPP_PP_CAT_I ## par
- #define RCPP_PP_CAT_I(a, b) RCPP_PP_CAT_II(a ## b)
- #define RCPP_PP_CAT_II(res) res

7.130.1 Macro Definition Documentation

7.130.1.1 RCPP_PP_CAT

```
#define RCPP_PP_CAT(
                a,
                b ) RCPP_PP_CAT_OO((a, b))
```

Definition at line 22 of file cat.hpp.

7.130.1.2 RCPP_PP_CAT_I

```
#define RCPP_PP_CAT_I(
                a,
                b ) RCPP_PP_CAT_II(a ## b)
```

Definition at line 29 of file cat.hpp.

7.130.1.3 RCPP_PP_CAT_II

```
#define RCPP_PP_CAT_II(
              res ) res
```

Definition at line 30 of file cat.hpp.

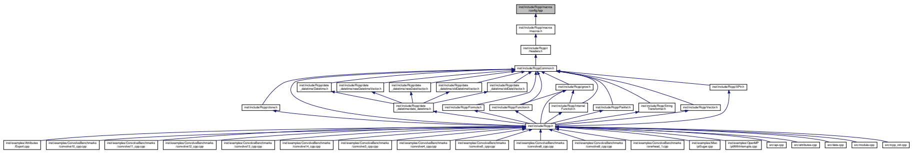
7.130.1.4 RCPP_PP_CAT_OO

```
#define RCPP_PP_CAT_OO(
              par ) RCPP_PP_CAT_I ## par
```

Definition at line 23 of file cat.hpp.

7.131 inst/include/Rcpp/macros/config.hpp File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/macros/config.hpp
inst/include/Rcpp/macros/macros.h
inst/include/Rcpp/r/headers.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define RCPP_PP_CONFIG_STRICT() 0x0001
- #define RCPP_PP_CONFIG_IDEAL() 0x0002
- #define RCPP_PP_CONFIG_MSVC() 0x0004
- #define RCPP_PP_CONFIG_MWCC() 0x0008
- #define RCPP_PP_CONFIG_BCC() 0x0010
- #define RCPP_PP_CONFIG_EDG() 0x0020
- #define RCPP_PP_CONFIG_DMC() 0x0040
- #define RCPP_PP_CONFIG_FLAGS() (RCPP_PP_CONFIG_STRICT())
- #define RCPP_PP_CONFIG_EXTENDED_LINE_INFO 0
- #define RCPP_PP_CONFIG_ERRORS 1

7.131.1 Macro Definition Documentation

7.131.1.1 RCPP_PP_CONFIG_BCC

```
#define RCPP_PP_CONFIG_BCC( ) 0x0010
```

Definition at line 22 of file config.hpp.

7.131.1.2 RCPP_PP_CONFIG_DMC

```
#define RCPP_PP_CONFIG_DMC( ) 0x0040
```

Definition at line 24 of file config.hpp.

7.131.1.3 RCPP_PP_CONFIG_EDG

```
#define RCPP_PP_CONFIG_EDG( ) 0x0020
```

Definition at line 23 of file config.hpp.

7.131.1.4 RCPP_PP_CONFIG_ERRORS

```
#define RCPP_PP_CONFIG_ERRORS 1
```

Definition at line 66 of file config.hpp.

7.131.1.5 RCPP_PP_CONFIG_EXTENDED_LINE_INFO

```
#define RCPP_PP_CONFIG_EXTENDED_LINE_INFO 0
```

Definition at line 57 of file config.hpp.

7.131.1.6 RCPP_PP_CONFIG_FLAGS

```
#define RCPP_PP_CONFIG_FLAGS( ) (RCPP_PP_CONFIG_STRICT())
```

Definition at line 50 of file config.hpp.

7.131.1.7 RCPP_PP_CONFIG_IDEAL

```
#define RCPP_PP_CONFIG_IDEAL( ) 0x0002
```

Definition at line 18 of file config.hpp.

7.131.1.8 RCPP_PP_CONFIG_MSVC

```
#define RCPP_PP_CONFIG_MSVC( ) 0x0004
```

Definition at line 20 of file config.hpp.

7.131.1.9 RCPP_PP_CONFIG_MWCC

```
#define RCPP_PP_CONFIG_MWCC( ) 0x0008
```

Definition at line 21 of file config.hpp.

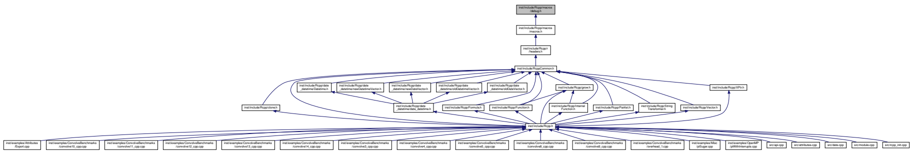
7.131.1.10 RCPP_PP_CONFIG_STRICT

```
#define RCPP_PP_CONFIG_STRICT( ) 0x0001
```

Definition at line 17 of file config.hpp.

7.132 inst/include/Rcpp/macros/debug.h File Reference

This graph shows which files directly or indirectly include this file:



inst/include/Rcpp/macros/debug.h
inst/include/Rcpp/macros/macros.h
inst/include/Rcpp/r/headers.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define RCPP_DEBUG_LEVEL 0
- #define RCPP_DEBUG_MODULE_LEVEL RCPP_DEBUG_LEVEL
- #define RCPP_DEBUG(MSG)
- #define RCPP_DEBUG_1(fmt, MSG)
- #define RCPP_DEBUG_2(fmt, M1, M2)
- #define RCPP_DEBUG_3(fmt, M1, M2, M3)
- #define RCPP_DEBUG_4(fmt, M1, M2, M3, M4)
- #define RCPP_DEBUG_5(fmt, M1, M2, M3, M4, M5)
- #define RCPP_DEBUG_MODULE(MSG)
- #define RCPP_DEBUG_MODULE_1(fmt, MSG)
- #define RCPP_DEBUG_MODULE_2(fmt, M1, M2)
- #define RCPP_DEBUG_MODULE_3(fmt, M1, M2, M3)
- #define RCPP_DEBUG_MODULE_4(fmt, M1, M2, M3, M4)
- #define RCPP_DEBUG_MODULE_5(fmt, M1, M2, M3, M4, M5)

7.132.1 Macro Definition Documentation

7.132.1.1 RCPP_DEBUG

```
#define RCPP_DEBUG(
              MSG )
```

Definition at line 43 of file debug.h.

7.132.1.2 RCPP_DEBUG_1

```
#define RCPP_DEBUG_1(
              fmt,
              MSG )
```

Definition at line 44 of file debug.h.

7.132.1.3 RCPP_DEBUG_2

```
#define RCPP_DEBUG_2(
              fmt,
              M1,
              M2 )
```

Definition at line 45 of file debug.h.

7.132.1.4 RCPP_DEBUG_3

```
#define RCPP_DEBUG_3(
              fmt,
              M1,
              M2,
              M3 )
```

Definition at line 46 of file debug.h.

7.132.1.5 RCPP_DEBUG_4

```
#define RCPP_DEBUG_4(
              fmt,
              M1,
              M2,
              M3,
              M4 )
```

Definition at line 47 of file debug.h.

7.132.1.6 RCPP_DEBUG_5

```
#define RCPP_DEBUG_5(
              fmt,
              M1,
              M2,
              M3,
              M4,
              M5 )
```

Definition at line 48 of file debug.h.

7.132.1.7 RCPP_DEBUG_LEVEL

```
#define RCPP_DEBUG_LEVEL 0
```

Definition at line 27 of file debug.h.

7.132.1.8 RCPP_DEBUG_MODULE

```
#define RCPP_DEBUG_MODULE(
              MSG )
```

Definition at line 101 of file debug.h.

7.132.1.9 RCPP_DEBUG_MODULE_1

```
#define RCPP_DEBUG_MODULE_1(
              fmt,
              MSG )
```

Definition at line 102 of file debug.h.

7.132.1.10 RCPP_DEBUG_MODULE_2

```
#define RCPP_DEBUG_MODULE_2(
              fmt,
              M1,
              M2 )
```

Definition at line 103 of file debug.h.

7.132.1.11 RCPP_DEBUG_MODULE_3

```
#define RCPP_DEBUG_MODULE_3(
              fmt,
              M1,
              M2,
              M3 )
```

Definition at line 104 of file debug.h.

7.132.1.12 RCPP_DEBUG_MODULE_4

```
#define RCPP_DEBUG_MODULE_4(
              fmt,
              M1,
              M2,
              M3,
              M4 )
```

Definition at line 105 of file debug.h.

7.132.1.13 RCPP_DEBUG_MODULE_5

```
#define RCPP_DEBUG_MODULE_5(
              fmt,
              M1,
              M2,
              M3,
              M4,
              M5 )
```

Definition at line 106 of file debug.h.

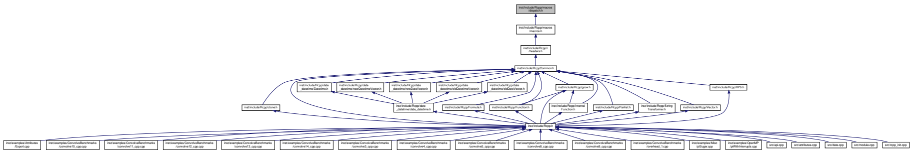
7.132.1.14 RCPP_DEBUG_MODULE_LEVEL

```
#define RCPP_DEBUG_MODULE_LEVEL RCPP_DEBUG_LEVEL
```

Definition at line 31 of file debug.h.

7.133 inst/include/Rcpp/macros/dispatch.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/macros/dispatch.h
inst/include/Rcpp/macros/macros.h
inst/include/Rcpp/r/headers.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define ___RCPP_HANDLE_CASE___(___RTYPE___, ___FUN___, ___OBJECT___, ___RCPPTYPE___)
- #define ___RCPP_RETURN___(__FUN__, __SEXP__, __RCPPTYPE__)
- #define RCPP_RETURN_VECTOR(_FUN_, _SEXP_) ___RCPP_RETURN___(_FUN_, _SEXP_, Vector)
- #define RCPP_RETURN_MATRIX(_FUN_, _SEXP_) ___RCPP_RETURN___(_FUN_, _SEXP_, Matrix)

7.133.1 Macro Definition Documentation

7.133.1.1 ___RCPP_HANDLE_CASE___

```
#define ___RCPP_HANDLE_CASE___(
                ___RTYPE___,
                ___FUN___,
                ___OBJECT___,
                ___RCPPTYPE___ )
```

Value:

```
  case ___RTYPE___:                                                             \
    return ___FUN___(::Rcpp::___RCPPTYPE___<___RTYPE___>(___OBJECT___));
```

Definition at line 82 of file dispatch.h.

7.133.1.2 ___RCPP_RETURN___

```
#define ___RCPP_RETURN___(
                __FUN__,
                __SEXP__,
                __RCPPTYPE__ )
```

Value:

```
  SEXP __TMP__ = __SEXP__;                                                      \
  switch (TYPEOF(__TMP__)) {                                                    \
    ___RCPP_HANDLE_CASE___(INTSXP, __FUN__, __TMP__, __RCPPTYPE__)              \
    ___RCPP_HANDLE_CASE___(REALSXP, __FUN__, __TMP__, __RCPPTYPE__)             \
    ___RCPP_HANDLE_CASE___(RAWSXP, __FUN__, __TMP__, __RCPPTYPE__)              \
    ___RCPP_HANDLE_CASE___(LGLSXP, __FUN__, __TMP__, __RCPPTYPE__)              \
    ___RCPP_HANDLE_CASE___(CPLXSXP, __FUN__, __TMP__, __RCPPTYPE__)             \
    ___RCPP_HANDLE_CASE___(STRSXP, __FUN__, __TMP__, __RCPPTYPE__)              \
    ___RCPP_HANDLE_CASE___(VECSXP, __FUN__, __TMP__, __RCPPTYPE__)              \
    ___RCPP_HANDLE_CASE___(EXPRSXP, __FUN__, __TMP__, __RCPPTYPE__)             \
  default:                                                                      \
    throw std::range_error("Not a vector");                                     \
  }
```

Definition at line 87 of file dispatch.h.

7.133.1.3 RCPP_RETURN_MATRIX

```
#define RCPP_RETURN_MATRIX(
            _FUN_,
            _SEXP_ )   ___RCPP_RETURN___(_FUN_, _SEXP_, Matrix)
```

Definition at line 104 of file dispatch.h.

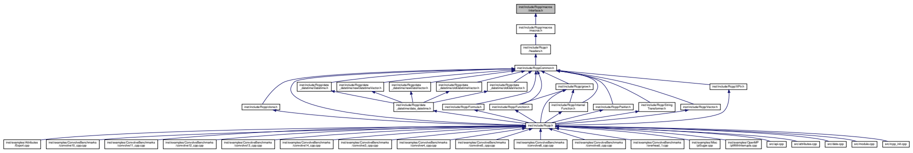
7.133.1.4 RCPP_RETURN_VECTOR

```
#define RCPP_RETURN_VECTOR(
            _FUN_,
            _SEXP_ )   ___RCPP_RETURN___(_FUN_, _SEXP_, Vector)
```

Definition at line 102 of file dispatch.h.

7.134 inst/include/Rcpp/macros/interface.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_GENERATE_CTOR_ASSIGN(__CLASS__)
- #define RCPP_CTOR_ASSIGN(__CLASS__)
- #define RCPP_CTOR_ASSIGN_WITH_BASE(__CLASS__)
- #define RCPP_API_CLASS(__CLASS__)

7.134.1 Macro Definition Documentation

7.134.1.1 RCPP_API_CLASS

```
#define RCPP_API_CLASS(
               __CLASS__ )
```

Value:

```
template < template <class> class StoragePolicy > class __CLASS__ :             \
    public StoragePolicy<__CLASS__<StoragePolicy> >,                            \
    public SlotProxyPolicy<__CLASS__<StoragePolicy> >,                          \
    public AttributeProxyPolicy<__CLASS__<StoragePolicy> >,                     \
    public RObjectMethods< __CLASS__<StoragePolicy> >
```

Definition at line 49 of file interface.h.

7.134.1.2 RCPP_CTOR_ASSIGN

```
#define RCPP_CTOR_ASSIGN(
               __CLASS__ )
```

Value:

```
__CLASS__( const __CLASS__& other ){                                            \
    Storage::copy__(other) ;                                                    \
}                                                                               \
__CLASS__& operator=(const __CLASS__& rhs) {                                    \
    return Storage::copy__(rhs) ;                                               \
}                                                                               \
template <typename Proxy>                                                       \
__CLASS__( const GenericProxy<Proxy>& proxy ){                                  \
    Storage::set__( proxy.get() ) ;                                             \
}
```

Definition at line 26 of file interface.h.

7.134.1.3 RCPP_CTOR_ASSIGN_WITH_BASE

```
#define RCPP_CTOR_ASSIGN_WITH_BASE(
               __CLASS__ )
```

Value:

```
  __CLASS__( const __CLASS__& other ) : Base(other) {                           \
}                                                                               \
__CLASS__& operator=(const __CLASS__& rhs) {                                    \
    return Storage::copy__(rhs) ;                                               \
}                                                                               \
template <typename Proxy>                                                       \
__CLASS__( const GenericProxy<Proxy>& proxy ){                                  \
    Storage::set__( proxy.get() ) ;                                             \
}
```

Definition at line 38 of file interface.h.

7.134.1.4 RCPP_GENERATE_CTOR_ASSIGN

```
#define RCPP_GENERATE_CTOR_ASSIGN(
              __CLASS__ )
```

Value:

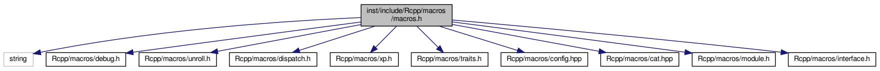
```
typedef StoragePolicy<__CLASS__> Storage ;                                    \
typedef AttributeProxyPolicy<__CLASS__> AttributePolicy ;                     \
RCPP_CTOR_ASSIGN(__CLASS__)
```

Definition at line 21 of file interface.h.

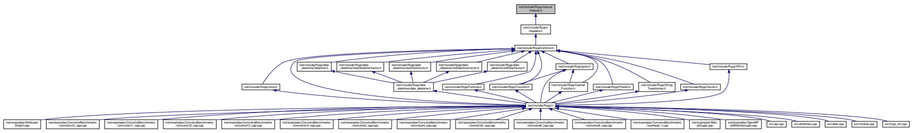
7.135 inst/include/Rcpp/macros/macros.h File Reference

```
#include <string>
#include <Rcpp/macros/debug.h>
#include <Rcpp/macros/unroll.h>
#include <Rcpp/macros/dispatch.h>
#include <Rcpp/macros/xp.h>
#include <Rcpp/macros/traits.h>
#include <Rcpp/macros/config.hpp>
#include <Rcpp/macros/cat.hpp>
#include <Rcpp/macros/module.h>
#include <Rcpp/macros/interface.h>
```

Include dependency graph for macros.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::internal::debug

Macros

- #define RCPP_DECORATE(__FUN__) __FUN__##__rcpp__wrapper__
- #define RCPP_GET_NAMES(x) Rf_getAttrib(x, R_NamesSymbol)
- #define RCPP_GET_CLASS(x) Rf_getAttrib(x, R_ClassSymbol)
- #define BEGIN_RCPP
- #define VOID_END_RCPP
- #define END_RCPP VOID_END_RCPP return R_NilValue;
- #define END_RCPP_RETURN_ERROR
- #define Rcpp_error(MESSAGE) throw Rcpp::exception(MESSAGE, __FILE__, __LINE__)

Functions

- std::string Rcpp::internal::debug::short_file_name (const char ∗file)

7.135.1 Macro Definition Documentation

7.135.1.1 BEGIN_RCPP

```
#define BEGIN_RCPP
```

Value:

```
    int rcpp_output_type = 0 ;                                                                   \
    int nprot = 0;                                                                               \
    (void)rcpp_output_type;                                                                      \
    SEXP rcpp_output_condition = R_NilValue ;                                                    \
    (void)rcpp_output_condition;                                                                 \
    static SEXP stop_sym = Rf_install("stop");                                                   \
    try {
```

Definition at line 49 of file macros.h.

7.135.1.2 END_RCPP

```
#define END_RCPP VOID_END_RCPP return R_NilValue;
```

Definition at line 99 of file macros.h.

7.135.1.3 END_RCPP_RETURN_ERROR

```
#define END_RCPP_RETURN_ERROR
```

Value:

```
  }                                                                              \
  catch (Rcpp::internal::InterruptedException &__ex__) {                         \
    return Rcpp::internal::interruptedError();                                   \
  }                                                                              \
  catch (Rcpp::LongjumpException& __ex__) {                                      \
    return Rcpp::internal::longjumpSentinel(__ex__.token);                       \
  }                                                                              \
  catch (std::exception &__ex__) {                                               \
    return exception_to_try_error(__ex__);                                       \
  }                                                                              \
  catch (...) {                                                                  \
    return string_to_try_error("c++ exception (unknown reason)");                \
  }                                                                              \
  UNPROTECT(nprot);                                                              \
  return R_NilValue;                                                             \
  (void) stop_sym;   /* never reached but suppresses warning */
```

Definition at line 106 of file macros.h.

7.135.1.4 RCPP_DECORATE

```
#define RCPP_DECORATE(
              __FUN__ ) __FUN__##__rcpp__wrapper__
```

Definition at line 44 of file macros.h.

7.135.1.5 Rcpp_error

```
#define Rcpp_error(
              MESSAGE ) throw Rcpp::exception(MESSAGE, __FILE__, __LINE__)
```

Definition at line 125 of file macros.h.

7.135.1.6 RCPP_GET_CLASS

```
#define RCPP_GET_CLASS(
              x ) Rf_getAttrib(x, R_ClassSymbol)
```

Definition at line 46 of file macros.h.

7.135.1.7 RCPP_GET_NAMES

```
#define RCPP_GET_NAMES(
              x ) Rf_getAttrib(x, R_NamesSymbol)
```

Definition at line 45 of file macros.h.

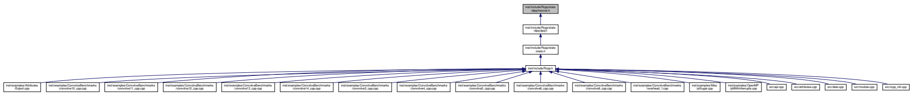
7.135.1.8 VOID_END_RCPP

```
#define VOID_END_RCPP
```

Definition at line 60 of file macros.h.

7.136 inst/include/Rcpp/stats/dpq/macros.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/dpq/macros.h
inst/include/Rcpp/stats/dpq/dpq.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

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)

7.136.1 Macro Definition Documentation

7.136.1.1 give_log

```
#define give_log log_p
```

Definition at line 24 of file macros.h.

7.136.1.2 R_D__0

```
#define R_D__0 (log_p ?  ML_NEGINF :  0.)  /∗ 0 ∗/
```

Definition at line 27 of file macros.h.

7.136.1.3 R_D__1

```
#define R_D__1 (log_p ?  0.  :  1.)  /∗ 1 ∗/
```

Definition at line 28 of file macros.h.

7.136.1.4 R_D_Clog

```
#define R_D_Clog(
              p ) (log_p ?  ::log1p(-(p)) :  (0.5 - (p) + 0.5)) /* [log](1-p) */
```

Definition at line 40 of file macros.h.

7.136.1.5 R_D_Cval

```
#define R_D_Cval(
              p ) (lower_tail ?  (0.5 - (p) + 0.5) :  (p)) /* 1 - p */
```

Definition at line 34 of file macros.h.

7.136.1.6 R_D_exp

```
#define R_D_exp(
              x ) (log_p ?  (x) :  ::exp(x)) /* exp(x) */
```

Definition at line 38 of file macros.h.

7.136.1.7 R_D_fexp

```
#define R_D_fexp(
              f,
              x ) (give_log ?  -0.5*log(f)+(x) :  exp(x)/sqrt(f))
```

Definition at line 116 of file macros.h.

7.136.1.8 R_D_forceint

```
#define R_D_forceint(
              x ) floor((x) + 0.5)
```

Definition at line 117 of file macros.h.

7.136.1.9 R_D_LExp

```
#define R_D_LExp(
              x ) (log_p ?  R_Log1_Exp(x) :  ::log1p(-x))
```

Definition at line 46 of file macros.h.

7.136.1.10 R_D_log

```
#define R_D_log(
              p ) (log_p ?  (p) :  ::log(p)) /* log(p) */
```

Definition at line 39 of file macros.h.

7.136.1.11 R_D_Lval

```
#define R_D_Lval(
              p ) (lower_tail ?  (p) :  (0.5 - (p) + 0.5)) /* p */
```

Definition at line 33 of file macros.h.

7.136.1.12 R_D_negInonint

```
#define R_D_negInonint(
              x ) (x < 0.  || R_D_nonint(x))
```

Definition at line 120 of file macros.h.

7.136.1.13 R_D_nonint

```
#define R_D_nonint(
              x ) (fabs((x) - floor((x)+0.5)) > 1e-7)
```

Definition at line 118 of file macros.h.

7.136.1.14 R_D_nonint_check

```
#define R_D_nonint_check(
              x )
```

Value:
```
   if(R_D_nonint(x)) {                                    \
       MATHLIB_WARNING("non-integer x = %f", x);       \
       return R_D__0;                                    \
   }
```

Definition at line 122 of file macros.h.

7.136.1.15 R_D_qIv

```
#define R_D_qIv(
              p ) (log_p ?  ::exp(p) :  (p)) /* p in qF(p,..)  */
```

Definition at line 37 of file macros.h.

7.136.1.16 R_D_val

```
#define R_D_val(
              x ) (log_p ?  ::log(x) :  (x)) /* x in pF(x,..)  */
```

Definition at line 36 of file macros.h.

7.136.1.17 R_DT_0

```
#define R_DT_0 (lower_tail ?  R_D__0 :  R_D__1) /* 0 */
```

Definition at line 29 of file macros.h.

7.136.1.18 R_DT_1

```
#define R_DT_1 (lower_tail ?  R_D__1 :  R_D__0) /* 1 */
```

Definition at line 30 of file macros.h.

7.136.1.19 R_DT_Cexp

```
#define R_DT_Cexp(
              x ) R_D_exp(R_D_Cval(x)) /* exp(1 - x) */
```

Definition at line 60 of file macros.h.

7.136.1.20 R_DT_CIv

```
#define R_DT_CIv(
              p )
```

Value:

```
(log_p ? (lower_tail ? -expm1(p) : ::exp(p)) \
: R_D_Cval(p))
```

Definition at line 56 of file macros.h.

7.136.1.21 R_DT_Clog

```
#define R_DT_Clog(
              p ) (lower_tail? R_D_LExp(p): R_D_log(p))/* log(1-p) in qF*/
```

Definition at line 63 of file macros.h.

7.136.1.22 R_DT_Cval

```
#define R_DT_Cval(
              x ) (lower_tail ? R_D_Clog(x) : R_D_val(x))
```

Definition at line 49 of file macros.h.

7.136.1.23 R_DT_exp

```
#define R_DT_exp(
              x ) R_D_exp(R_D_Lval(x)) /* exp(x) */
```

Definition at line 59 of file macros.h.

7.136.1.24 R_DT_log

```
#define R_DT_log(
              p ) (lower_tail? R_D_log(p) : R_D_LExp(p))/* log(p) in qF */
```

Definition at line 62 of file macros.h.

7.136.1.25 R_DT_Log

```
#define R_DT_Log(
              p ) (lower_tail? (p) : R_Log1_Exp(p))
```

Definition at line 64 of file macros.h.

7.136.1.26 R_DT_qIv

```
#define R_DT_qIv(
              p )
```

Value:

```
(log_p ? (lower_tail ? ::exp(p) : - ::expm1(p)) \
: R_D_Lval(p))
```

Definition at line 52 of file macros.h.

7.136.1.27 R_DT_val

```
#define R_DT_val(
              x ) (lower_tail ? R_D_val(x) : R_D_Clog(x))
```

Definition at line 48 of file macros.h.

7.136.1.28 R_Log1_Exp

```
#define R_Log1_Exp(
              x ) ((x) > -M_LN2 ? ::log(-::expm1(x)) : ::log1p(-::exp(x)))
```

Definition at line 43 of file macros.h.

7.136.1.29 R_P_bounds_01

```
#define R_P_bounds_01(
               x,
               x_min,
               x_max )
```

Value:

```
    if(x <= x_min) return R_DT_0;          \
    if(x >= x_max) return R_DT_1
```

Definition at line 102 of file macros.h.

7.136.1.30 R_P_bounds_Inf_01

```
#define R_P_bounds_Inf_01(
               x )
```

Value:

```
    if(!R_FINITE(x)) {                           \
        if (x > 0) return R_DT_1;          \
        /* x < 0 */return R_DT_0;          \
    }
```

Definition at line 107 of file macros.h.

7.136.1.31 R_Q_P01_boundaries

```
#define R_Q_P01_boundaries(
               p,
               _LEFT_,
               _RIGHT_ )
```

Value:

```
    if (log_p) {                                         \
        if(p > 0)                                        \
            return R_NaN ;                               \
        if(p == 0) /* upper bound*/                      \
            return lower_tail ? _RIGHT_ : _LEFT_;        \
        if(p == ML_NEGINF)                              \
            return lower_tail ? _LEFT_ : _RIGHT_;        \
    }                                                    \
    else { /* !log_p */                                  \
        if(p < 0 || p > 1)                               \
            return R_NaN ;                               \
        if(p == 0)                                       \
            return lower_tail ? _LEFT_ : _RIGHT_;        \
        if(p == 1)                                       \
            return lower_tail ? _RIGHT_ : _LEFT_;        \
    }
```

Definition at line 84 of file macros.h.

7.136.1.32 R_Q_P01_check

```
#define R_Q_P01_check(
                p )
```

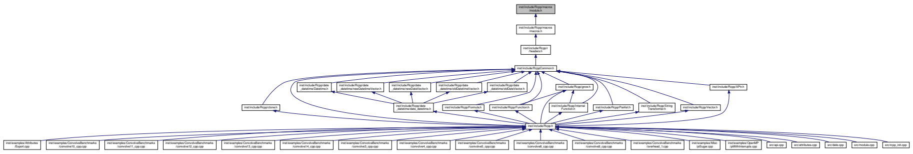
Value:

```
    if ((log_p  && p > 0) ||                        \
        (!log_p && (p < 0 || p > 1)) )              \
        return R_NaN
```

Definition at line 68 of file macros.h.

7.137 inst/include/Rcpp/macros/module.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_EXPOSED_AS(CLASS)
- #define RCPP_EXPOSED_WRAP(CLASS) namespace Rcpp{ namespace traits{ template<> struct wrap_↩ type_traits< CLASS >{typedef wrap_type_module_object_tag wrap_category ; } ; }}
- #define RCPP_EXPOSED_CLASS_NODECL(CLASS)
- #define RCPP_EXPOSED_CLASS(CLASS)
- #define RCPP_EXPOSED_ENUM_AS(CLASS) namespace Rcpp{ namespace traits{ template<> struct r_↩ type_traits< CLASS >{ typedef r_type_enum_tag r_category ; } ; }}
- #define RCPP_EXPOSED_ENUM_WRAP(CLASS) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS >{typedef wrap_type_enum_tag wrap_category ; } ; }}
- #define RCPP_EXPOSED_ENUM_NODECL(CLASS)
- #define RCPP_EXPOSED_ENUM(CLASS)

7.137.1 Macro Definition Documentation

7.137.1.1 RCPP_EXPOSED_AS

```
#define RCPP_EXPOSED_AS(
                CLASS )
```

This macros should be used by packages using modules when a type is used as a parameter of a function or method exposed by modules. This defines the necessary trait that makes the class as<>'able

Definition at line 29 of file module.h.

7.137.1.2 RCPP_EXPOSED_CLASS

```
#define RCPP_EXPOSED_CLASS(
              CLASS )
```

Value:

```
class CLASS;                     \
RCPP_EXPOSED_CLASS_NODECL(CLASS)
```

Definition at line 69 of file module.h.

7.137.1.3 RCPP_EXPOSED_CLASS_NODECL

```
#define RCPP_EXPOSED_CLASS_NODECL(
              CLASS )
```

Value:

```
RCPP_EXPOSED_AS(CLASS)           \
RCPP_EXPOSED_WRAP(CLASS)
```

Definition at line 65 of file module.h.

7.137.1.4 RCPP_EXPOSED_ENUM

```
#define RCPP_EXPOSED_ENUM(
              CLASS )
```

Value:

```
class CLASS;                     \
RCPP_EXPOSED_ENUM_NODECL(CLASS)
```

Definition at line 83 of file module.h.

7.137.1.5 RCPP_EXPOSED_ENUM_AS

```
#define RCPP_EXPOSED_ENUM_AS(
              CLASS ) namespace Rcpp{ namespace traits{ template<> struct r_type_traits< CLASS
>{ typedef r_type_enum_tag r_category ; } ; }}
```

handling enums: TODO use is_enum from C++11 or boost to have those automatic

Definition at line 76 of file module.h.

7.137.1.6 RCPP_EXPOSED_ENUM_NODECL

```
#define RCPP_EXPOSED_ENUM_NODECL(
                CLASS )
```

Value:

```
  RCPP_EXPOSED_ENUM_AS(CLASS)           \
  RCPP_EXPOSED_ENUM_WRAP(CLASS)
```

Definition at line 79 of file module.h.

7.137.1.7 RCPP_EXPOSED_ENUM_WRAP

```
#define RCPP_EXPOSED_ENUM_WRAP(
                CLASS ) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS
>{typedef wrap_type_enum_tag wrap_category ; } ; }}
```

Definition at line 77 of file module.h.

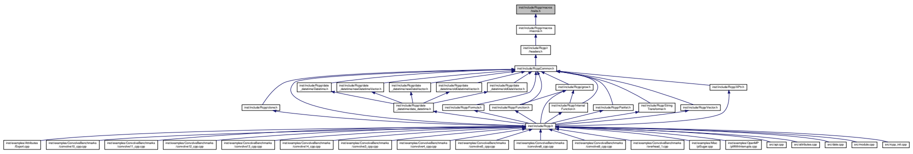
7.137.1.8 RCPP_EXPOSED_WRAP

```
#define RCPP_EXPOSED_WRAP(
                CLASS ) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS
>{typedef wrap_type_module_object_tag wrap_category ; } ; }}
```

Definition at line 63 of file module.h.

7.138 inst/include/Rcpp/macros/traits.h File Reference

This graph shows which files directly or indirectly include this file:



inst/include/Rcpp/macros/traits.h
inst/include/Rcpp/macros/macros.h
inst/include/Rcpp/r/headers.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define RCPP_TRAITS(__CLASS__, __SEXPTYPE__)
- #define RCPP_ENUM_TRAITS(__ENUM__) RCPP_TRAITS(__ENUM__,INTSXP)

7.138.1 Macro Definition Documentation

7.138.1.1 RCPP_ENUM_TRAITS

```
#define RCPP_ENUM_TRAITS(
              __ENUM__ ) RCPP_TRAITS(__ENUM__,INTSXP)
```

Definition at line 40 of file traits.h.

7.138.1.2 RCPP_TRAITS

```
#define RCPP_TRAITS(
              __CLASS__,
              __SEXPTYPE__ )
```

Value:

```
namespace Rcpp{ namespace traits {                                  \
template<> struct r_type_traits< __CLASS__ >{                         \
        typedef r_type_primitive_tag r_category ;                       \
} ;                                                                 \
template<> struct r_type_traits< std::pair< std::string , __CLASS__ > >{   \
        typedef r_type_pairstring_primitive_tag r_category ;            \
} ;                                                                 \
template<> struct wrap_type_traits< __CLASS__ >{                      \
        typedef wrap_type_primitive_tag wrap_category ;                 \
} ;                                                                 \
template<> struct r_sexptype_traits< __CLASS__ >{                     \
        enum{ rtype = __SEXPTYPE__ } ;                                  \
} ;                                                                 \
} }
```

Definition at line 25 of file traits.h.

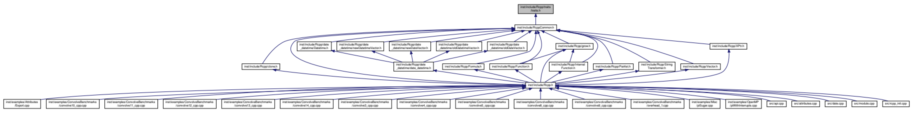
7.139 inst/include/Rcpp/traits/traits.h File Reference

```
#include <Rcpp/traits/integral_constant.h>
#include <Rcpp/traits/same_type.h>
#include <Rcpp/traits/enable_if.h>
#include <Rcpp/traits/is_wide_string.h>
#include <Rcpp/traits/is_arithmetic.h>
#include <Rcpp/traits/char_type.h>
#include <Rcpp/traits/named_object.h>
#include <Rcpp/traits/is_convertible.h>
#include <Rcpp/traits/has_iterator.h>
#include <Rcpp/traits/expands_to_logical.h>
#include <Rcpp/traits/matrix_interface.h>
#include <Rcpp/traits/is_sugar_expression.h>
#include <Rcpp/traits/is_eigen_base.h>
```

```
#include <Rcpp/traits/has_na.h>
#include <Rcpp/traits/storage_type.h>
#include <Rcpp/traits/r_sexptype_traits.h>
#include <Rcpp/traits/r_type_traits.h>
#include <Rcpp/traits/un_pointer.h>
#include <Rcpp/traits/is_pointer.h>
#include <Rcpp/traits/wrap_type_traits.h>
#include <Rcpp/traits/longlong.h>
#include <Rcpp/traits/module_wrap_traits.h>
#include <Rcpp/traits/is_na.h>
#include <Rcpp/traits/is_finite.h>
#include <Rcpp/traits/is_infinite.h>
#include <Rcpp/traits/is_nan.h>
#include <Rcpp/traits/is_bool.h>
#include <Rcpp/traits/if_.h>
#include <Rcpp/traits/get_na.h>
#include <Rcpp/traits/is_trivial.h>
#include <Rcpp/traits/init_type.h>
#include <Rcpp/traits/is_const.h>
#include <Rcpp/traits/is_reference.h>
#include <Rcpp/traits/remove_const.h>
#include <Rcpp/traits/remove_reference.h>
#include <Rcpp/traits/remove_const_and_reference.h>
#include <Rcpp/traits/result_of.h>
#include <Rcpp/traits/is_module_object.h>
#include <Rcpp/traits/is_primitive.h>
#include <Rcpp/traits/one_type.h>
```

Include dependency graph for traits.h:

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::identity< T >
- struct Rcpp::traits::int2type< I >

Namespaces

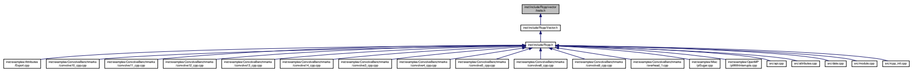
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.140 inst/include/Rcpp/vector/traits.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >
- class Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >
- struct Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy >
- struct Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >

Namespaces

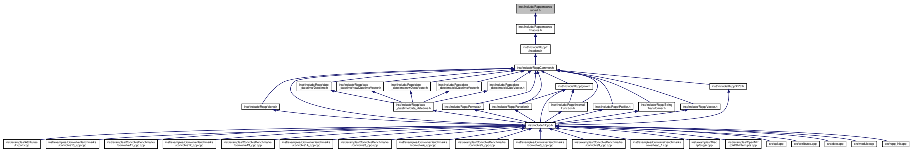
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.141 inst/include/Rcpp/macros/unroll.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_LOOP_UNROLL_PTR(TARGET, SOURCE)
- #define RCPP_LOOP_UNROLL(TARGET, SOURCE)
- #define RCPP_LOOP_UNROLL_LHSFUN(TARGET, FUN, SOURCE)

7.141.1 Macro Definition Documentation

7.141.1.1 RCPP_LOOP_UNROLL

```
#define RCPP_LOOP_UNROLL(
                TARGET,
                SOURCE )
```

Value:

```
R_xlen_t __trip_count = n » 2 ;                        \
R_xlen_t i = 0 ;                                        \
for ( ; __trip_count > 0 ; --__trip_count) {            \
        TARGET[i] = SOURCE[i] ; i++ ;                   \
        TARGET[i] = SOURCE[i] ; i++ ;                   \
        TARGET[i] = SOURCE[i] ; i++ ;                   \
        TARGET[i] = SOURCE[i] ; i++ ;                   \
}                                                       \
switch (n - i){                                         \
  case 3:                                               \
      TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 2:                                               \
      TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 1:                                               \
      TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 0:                                               \
  default:                                              \
      {}                                                \
}
```

Definition at line 47 of file unroll.h.

7.141.1.2 RCPP_LOOP_UNROLL_LHSFUN

```
#define RCPP_LOOP_UNROLL_LHSFUN(
                TARGET,
                FUN,
                SOURCE )
```

Value:

```
R_xlen_t __trip_count = n » 2 ;                             \
R_xlen_t i = 0 ;                                             \
for ( ; __trip_count > 0 ; --__trip_count) {                 \
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;                       \
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;                       \
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;                       \
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;                       \
}                                                            \
switch (n - i){                                              \
  case 3:                                                    \
      TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 2:                                                    \
      TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 1:                                                    \
      TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */ \
  case 0:                                                    \
  default:                                                   \
      {}                                                     \
}
```

Definition at line 68 of file unroll.h.

7.141.1.3 RCPP_LOOP_UNROLL_PTR

```
#define RCPP_LOOP_UNROLL_PTR(
               TARGET,
               SOURCE )
```

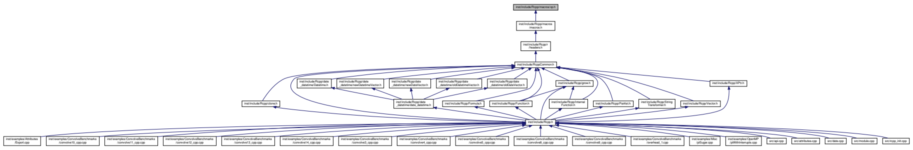
Value:

```
R_xlen_t __trip_count = n » 2 ;                   \
R_xlen_t i = 0 ;                                   \
for ( ; __trip_count > 0 ; --__trip_count) {       \
        *TARGET++ = SOURCE[i++] ;                      \
        *TARGET++ = SOURCE[i++] ;                      \
        *TARGET++ = SOURCE[i++] ;                      \
        *TARGET++ = SOURCE[i++] ;                      \
}                                                  \
switch (n - i){                                    \
  case 3:                                          \
    *TARGET++ = SOURCE[i++] ;                      \
  case 2:                                          \
    *TARGET++ = SOURCE[i++] ;                      \
  case 1:                                          \
    *TARGET++ = SOURCE[i++] ;                      \
  case 0:                                          \
  default:                                         \
      {}                                           \
}
```

Definition at line 25 of file unroll.h.

7.142 inst/include/Rcpp/macros/xp.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_XP_FIELD_GET(__NAME__, __CLASS__, __FIELD__)
- #define RCPP_XP_FIELD_SET(__NAME__, __CLASS__, __FIELD__)
- #define RCPP_XP_FIELD(__PREFIX__, __CLASS__, __FIELD__)

7.142.1 Macro Definition Documentation

7.142.1.1 RCPP_XP_FIELD

```
#define RCPP_XP_FIELD(
              __PREFIX__,
              __CLASS__,
              __FIELD__ )
```

Value:

```
RCPP_XP_FIELD_GET( RCPP_PP_CAT(__PREFIX__,_get), __CLASS__, __FIELD__ )    \
RCPP_XP_FIELD_SET( RCPP_PP_CAT(__PREFIX__,_set), __CLASS__, __FIELD__ )
```

Definition at line 61 of file xp.h.

7.142.1.2 RCPP_XP_FIELD_GET

```
#define RCPP_XP_FIELD_GET(
              __NAME__,
              __CLASS__,
              __FIELD__ )
```

Value:

```
extern "C" SEXP RCPP_PP_CAT(__NAME__,__rcpp_info__)(){         \
        using Rcpp::_ ;                                             \
        Rcpp::List info = Rcpp::List::create(                       \
        _["class"]  = #__CLASS__  ,                             \
        _["field"]  = #__FIELD__                              \
        )   ;                                                   \
    info.attr( "class" ) = "rcppxpfieldgetinfo" ;               \
    return info   ;                                             \
}                                                               \
extern "C" SEXP __NAME__( SEXP xp ){                            \
        BEGIN_RCPP                                                  \
            SEXP res = R_NilValue ;                                 \
                ::Rcpp::XPtr< __CLASS__ > ptr(xp) ;                     \
                res = ::Rcpp::wrap( ptr->__FIELD__ ) ;                  \
                return res ;                                            \
        END_RCPP                                                    \
}
```

Definition at line 25 of file xp.h.

7.142.1.3 RCPP_XP_FIELD_SET

```
#define RCPP_XP_FIELD_SET(
              __NAME__,
              __CLASS__,
              __FIELD__ )
```

Value:

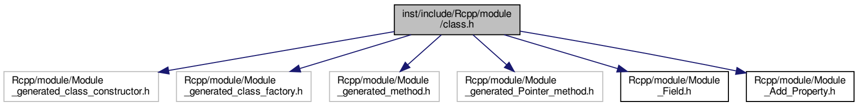
```
extern "C" SEXP RCPP_PP_CAT(__NAME__,__rcpp_info__)(){         \
        using Rcpp::_ ;                                             \
        Rcpp::List info = Rcpp::List::create(                       \
        _["class"]  = #__CLASS__  ,                             \
        _["field"]  = #__FIELD__                              \
        )   ;                                                   \
    info.attr( "class" ) = "rcppxpfieldsetinfo" ;               \
    return info   ;                                             \
}                                                               \
extern "C" SEXP __NAME__( SEXP xp, SEXP value ){                \
        BEGIN_RCPP                                                  \
                ::Rcpp::XPtr< __CLASS__ > ptr(xp) ;                     \
                ptr->__FIELD__ = ::Rcpp::internal::converter(value) ;  \
        END_RCPP                                                    \
}
```

Definition at line 44 of file xp.h.

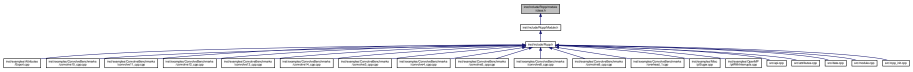
7.143 inst/include/Rcpp/module/class.h File Reference

```
#include <Rcpp/module/Module_generated_class_constructor.h>
#include <Rcpp/module/Module_generated_class_factory.h>
#include <Rcpp/module/Module_generated_method.h>
#include <Rcpp/module/Module_generated_Pointer_method.h>
#include <Rcpp/module/Module_Field.h>
#include <Rcpp/module/Module_Add_Property.h>
```

Include dependency graph for class.h:



This graph shows which files directly or indirectly include this file:

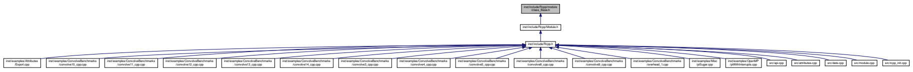


Classes

- class class_< Class >

7.144 inst/include/Rcpp/module/class_Base.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::class_Base

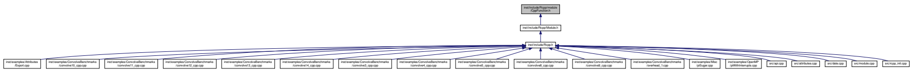
Namespaces

- Rcpp

  Rcpp API.

7.145 inst/include/Rcpp/module/CppFunction.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/module/CppFunction.h
inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::CppFunctionBase
- class Rcpp::CppFunction

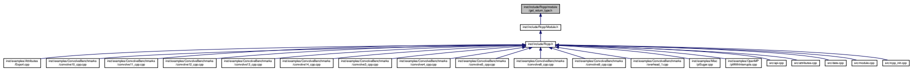
Namespaces

- Rcpp

  Rcpp API.

7.146 inst/include/Rcpp/module/get_return_type.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/module/get_return_type.h
inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::void_type

Namespaces

- Rcpp

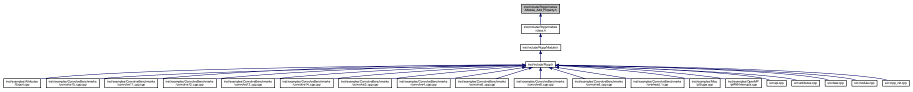
  Rcpp API.

Functions

- template<typename RESULT_TYPE >
  std::string Rcpp::get_return_type_dispatch (Rcpp::traits::false_type)
- template<typename RESULT_TYPE >
  std::string Rcpp::get_return_type_dispatch (Rcpp::traits::true_type)
- template<typename RESULT_TYPE >
  std::string Rcpp::get_return_type ()
- template<> std::string Rcpp::get_return_type< void_type > ()
- template<> std::string Rcpp::get_return_type< SEXP > ()
- template<> std::string Rcpp::get_return_type< Rcpp::IntegerVector > ()
- template<> std::string Rcpp::get_return_type< Rcpp::NumericVector > ()
- template<> std::string Rcpp::get_return_type< Rcpp::RawVector > ()
- template<> std::string Rcpp::get_return_type< Rcpp::ExpressionVector > ()
- template<> std::string Rcpp::get_return_type< Rcpp::List > ()
- template<> std::string Rcpp::get_return_type< Rcpp::CharacterVector > ()

7.147 inst/include/Rcpp/module/Module_Add_Property.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- template<typename PROP >
  self & property (const char ∗name_, PROP(Class::∗GetMethod)(void), const char ∗docstring=0)
- template<typename PROP >
  self & property (const char ∗name_, PROP(∗GetMethod)(Class ∗), const char ∗docstring)
- template<typename PROP >
  self & property (const char ∗name_, PROP(Class::∗GetMethod)(void), void(Class::∗SetMethod)(PROP), const char ∗docstring=0)
- template<typename PROP >
  self & property (const char ∗name_, PROP(Class::∗GetMethod)(void), void(∗SetMethod)(Class ∗, PROP), const char ∗docstring=0)
- template<typename PROP >
  self & property (const char ∗name_, PROP(∗GetMethod)(Class ∗), void(Class::∗SetMethod)(PROP), const char ∗docstring=0)
- template<typename PROP >
  self & property (const char ∗name_, PROP(∗GetMethod)(Class ∗), void(∗SetMethod)(Class ∗, PROP), const char ∗docstring=0)

7.147.1 Function Documentation

7.147.1.1 property() [1/6]

```
template<typename PROP >
self& property (
            const char * name_,
            PROP(*)(Class *) GetMethod,
            const char * docstring )
```

Definition at line 38 of file Module_Add_Property.h.

7.147.1.2 property() [2/6]

```
template<typename PROP >
self& property (
            const char * name_,
            PROP(*)(Class *) GetMethod,
            void(*)(Class *, PROP) SetMethod,
            const char * docstring = 0 )
```

Definition at line 89 of file Module_Add_Property.h.

7.147.1.3 property() [3/6]

```
template<typename PROP >
self& property (
            const char * name_,
            PROP(*)(Class *) GetMethod,
            void(Class::*)(PROP) SetMethod,
            const char * docstring = 0 )
```

Definition at line 81 of file Module_Add_Property.h.

7.147.1.4 property() [4/6]

```
template<typename PROP >
self & property (
            const char * name_,
            PROP(Class::*)(void) GetMethod,
            const char * docstring = 0 )
```

Definition at line 26 of file Module_Add_Property.h.

7.147.1.5 property() [5/6]

```
template<typename PROP >
self & property (
             const char * name_,
             PROP(Class::*)(void) GetMethod,
             void(*)(Class *, PROP) SetMethod,
             const char * docstring = 0 )
```

Definition at line 63 of file Module_Add_Property.h.

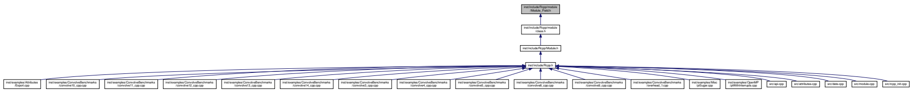
7.147.1.6 property() [6/6]

```
template<typename PROP >
self & property (
             const char * name_,
             PROP(Class::*)(void) GetMethod,
             void(Class::*)(PROP) SetMethod,
             const char * docstring = 0 )
```

Definition at line 45 of file Module_Add_Property.h.

7.148 inst/include/Rcpp/module/Module_Field.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/module/Module_Field.h
inst/include/Rcpp/module/class.h
inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class CppProperty_Getter_Setter< PROP >
- class CppProperty_Getter< PROP >

Functions

- template<typename T >
  self & field (const char *name_, T Class::*ptr, const char *docstring=0)
- template<typename T >
  self & field_readonly (const char *name_, T Class::*ptr, const char *docstring=0)

7.148.1 Function Documentation

7.148.1.1 field()

```
template<typename T >
self& field (
             const char * name_,
             T Class::* ptr,
             const char * docstring = 0 )
```

Definition at line 68 of file Module_Field.h.

Referenced by Rcpp::S4_CppConstructor< Class >::S4_CppConstructor(), Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods(), and Rcpp::S4_field< Class >::S4_field().

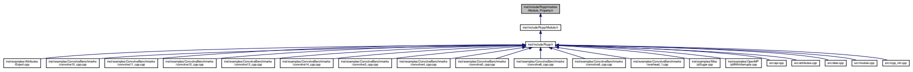
7.148.1.2 field_readonly()

```
template<typename T >
self& field_readonly (
             const char * name_,
             T Class::* ptr,
             const char * docstring = 0 )
```

Definition at line 76 of file Module_Field.h.

7.149 inst/include/Rcpp/module/Module_Property.h File Reference

This graph shows which files directly or indirectly include this file:

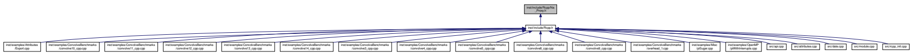

inst/include/Rcpp/module/Module_Property.h
inst/include/Rcpp/Module.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class CppProperty_GetMethod< Class, PROP >
- class CppProperty_GetConstMethod< Class, PROP >
- class CppProperty_GetPointerMethod< Class, PROP >
- class CppProperty_GetMethod_SetMethod< Class, PROP >
- class CppProperty_GetConstMethod_SetMethod< Class, PROP >
- class CppProperty_GetMethod_SetPointer< Class, PROP >
- class CppProperty_GetConstMethod_SetPointer< Class, PROP >
- class CppProperty_GetPointer_SetMethod< Class, PROP >
- class CppProperty_GetPointer_SetPointer< Class, PROP >

7.150 inst/include/Rcpp/Na_Proxy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Na_Proxy.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::Na_Proxy

Namespaces

- Rcpp

  Rcpp API.

Functions

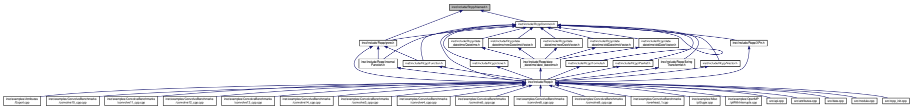
- LogicalVector Rcpp::shush_about_NA ()

Variables

- static Na_Proxy Rcpp::NA

7.151 inst/include/Rcpp/Named.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Named.h
inst/include/RcppCommon.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::Argument
- class Rcpp::internal::NamedPlaceHolder

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Functions

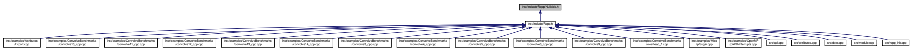
- Argument Rcpp::Named (const std::string &name)
- template<typename T >
  traits::named_object< T > Rcpp::Named (const std::string &name, const T &o)

Variables

- static internal::NamedPlaceHolder Rcpp::_

7.152 inst/include/Rcpp/Nullable.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Nullable.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::Nullable< T >

Namespaces

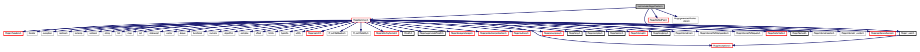
- Rcpp

  Rcpp API.

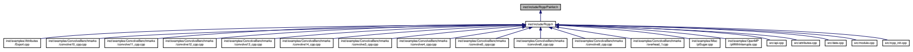
7.153 inst/include/Rcpp/Pairlist.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/DottedPair.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/generated/Pairlist__ctors.h>
```

Include dependency graph for Pairlist.h:



This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Pairlist.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef Pairlist_Impl< PreserveStorage > Rcpp::Pairlist

Functions

- Rcpp::RCPP_API_CLASS (Pairlist_Impl)
- Rcpp::Pairlist_Impl ()
- Rcpp::Pairlist_Impl (SEXP x)
- void Rcpp::update (SEXP)

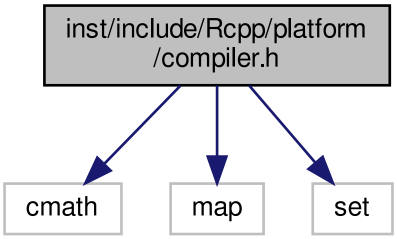
Variables

- public Rcpp::DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >
- public public Rcpp::DottedPairImpl< Pairlist_Impl< StoragePolicy > >

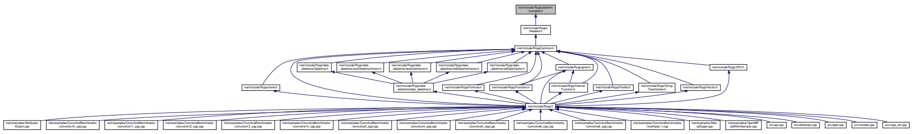
7.154 inst/include/Rcpp/platform/compiler.h File Reference

```
#include <cmath>
#include <map>
#include <set>
```

Include dependency graph for compiler.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define RCPP_USING_MAP
- #define RCPP_UNORDERED_MAP std::map
- #define RCPP_USING_SET
- #define RCPP_UNORDERED_SET std::set

7.154.1 Macro Definition Documentation

7.154.1.1 RCPP_UNORDERED_MAP

```
#define RCPP_UNORDERED_MAP std::map
```

Definition at line 178 of file compiler.h.

7.154.1.2 RCPP_UNORDERED_SET

```
#define RCPP_UNORDERED_SET std::set
```

Definition at line 187 of file compiler.h.

7.154.1.3 RCPP_USING_MAP

```
#define RCPP_USING_MAP
```

Definition at line 177 of file compiler.h.

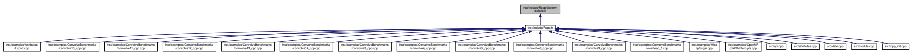
7.154.1.4 RCPP_USING_SET

```
#define RCPP_USING_SET
```

Definition at line 186 of file compiler.h.

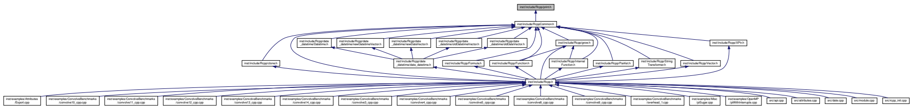
7.155 inst/include/Rcpp/platform/solaris.h File Reference

This graph shows which files directly or indirectly include this file:



7.156 inst/include/Rcpp/print.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

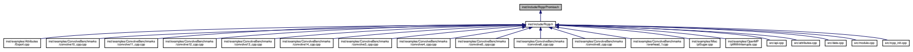
  Rcpp API.

Functions

- void Rcpp::print (SEXP s)
- void Rcpp::warningcall (SEXP call, const std::string &s)

7.157 inst/include/Rcpp/Promise.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

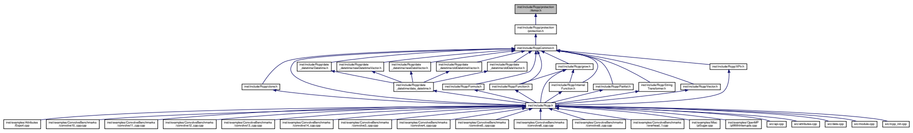
- typedef Promise_Impl< PreserveStorage > Rcpp::Promise

Functions

- Rcpp::RCPP_API_CLASS (Promise_Impl)

7.158 inst/include/Rcpp/protection/Armor.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Armor< T >

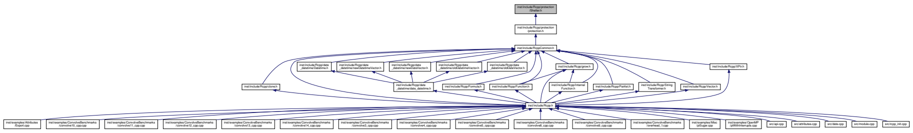
Namespaces

- Rcpp

  Rcpp API.

7.159 inst/include/Rcpp/protection/Shelter.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Shelter< T >

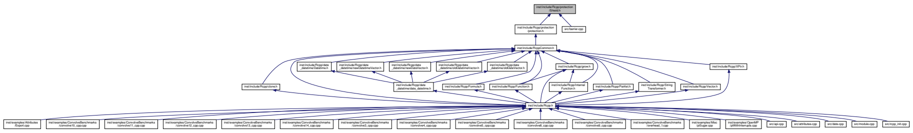
Namespaces

- Rcpp

    Rcpp API.

7.160 inst/include/Rcpp/protection/Shield.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Shield< T >

Namespaces

- Rcpp

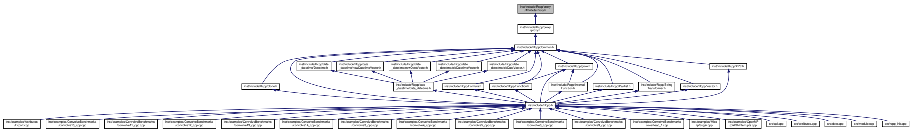
    Rcpp API.

Functions

- SEXP Rcpp::Rcpp_protect (SEXP x)
- void Rcpp::Rcpp_unprotect (int i)

7.161 inst/include/Rcpp/proxy/AttributeProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::AttributeProxyPolicy< CLASS >
- class Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy
- class Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy

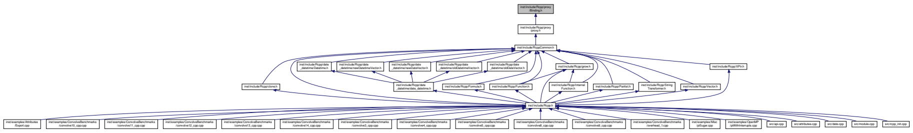
Namespaces

- Rcpp

  Rcpp API.

7.162 inst/include/Rcpp/proxy/Binding.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::BindingPolicy< EnvironmentClass >
- class Rcpp::BindingPolicy< EnvironmentClass >::Binding
- class Rcpp::BindingPolicy< EnvironmentClass >::const_Binding

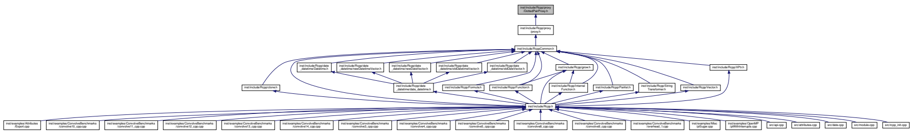
Namespaces

- Rcpp

  Rcpp API.

7.163 inst/include/Rcpp/proxy/DottedPairProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::DottedPairProxyPolicy< CLASS >
- class Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy
- class Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy

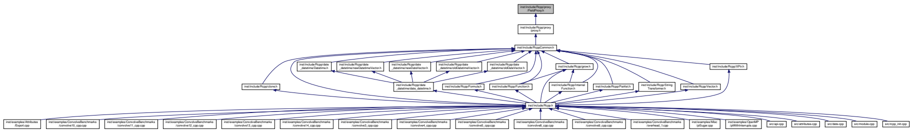
Namespaces

- Rcpp

  Rcpp API.

7.164 inst/include/Rcpp/proxy/FieldProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::FieldProxyPolicy< CLASS >
- class Rcpp::FieldProxyPolicy< CLASS >::FieldProxy
- class Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy

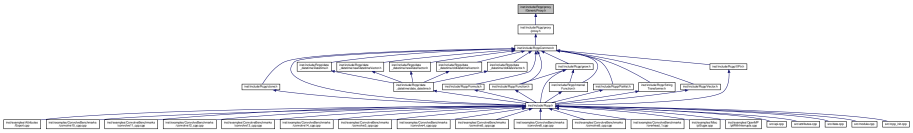
Namespaces

- Rcpp

  Rcpp API.

7.165 inst/include/Rcpp/proxy/GenericProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::GenericProxy< Proxy >

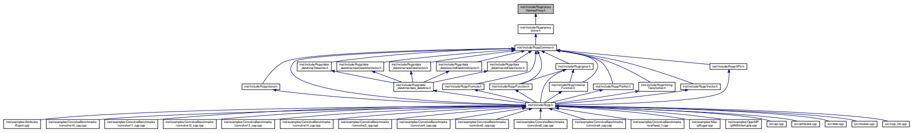
Namespaces

- Rcpp

  Rcpp API.

7.166 inst/include/Rcpp/proxy/NamesProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::NamesProxyPolicy< CLASS >
- class Rcpp::NamesProxyPolicy< CLASS >::NamesProxy
- class Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy

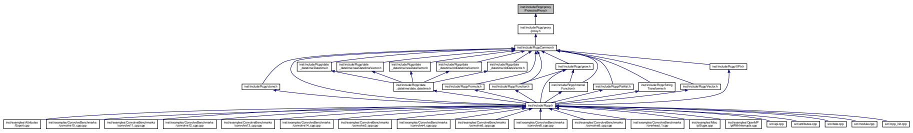
Namespaces

- Rcpp

  Rcpp API.

7.167 inst/include/Rcpp/proxy/ProtectedProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::ProtectedProxyPolicy< XPtrClass >
- class Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy
- class Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy

Namespaces

- Rcpp

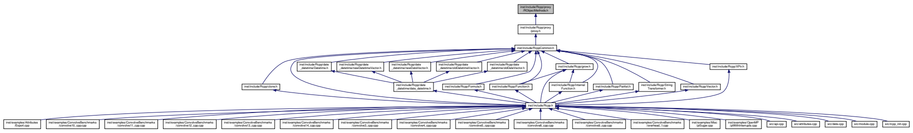
  Rcpp API.

Functions

- template< typename T >
  T Rcpp::as (SEXP x)

7.168 inst/include/Rcpp/proxy/RObjectMethods.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::RObjectMethods< Class >

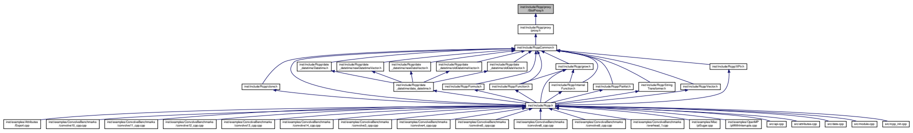
Namespaces

- Rcpp

  Rcpp API.

7.169 inst/include/Rcpp/proxy/SlotProxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::SlotProxyPolicy< CLASS >
- class Rcpp::SlotProxyPolicy< CLASS >::SlotProxy
- class Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy

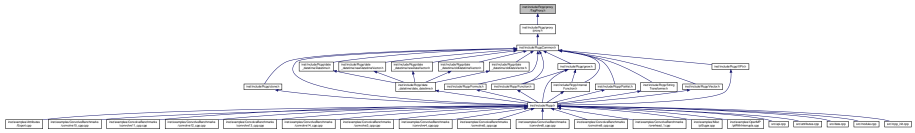
Namespaces

- Rcpp

  Rcpp API.

7.170 inst/include/Rcpp/proxy/TagProxy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/proxy/TagProxy.h
inst/include/Rcpp/proxy/proxy.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::TagProxyPolicy< XPtrClass >
- class Rcpp::TagProxyPolicy< XPtrClass >::TagProxy
- class Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy

Namespaces

- Rcpp

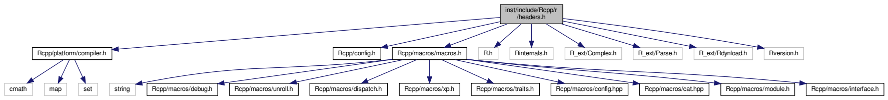
  Rcpp API.

7.171 inst/include/Rcpp/r/headers.h File Reference

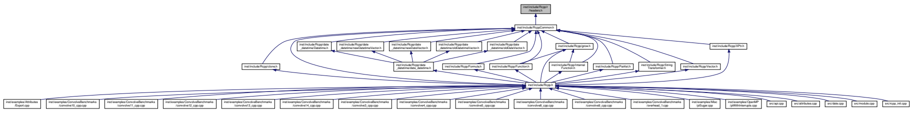
```
#include <Rcpp/platform/compiler.h>
#include <Rcpp/config.h>
#include <Rcpp/macros/macros.h>
#include <R.h>
#include <Rinternals.h>
#include <R_ext/Complex.h>
#include <R_ext/Parse.h>
#include <R_ext/Rdynload.h>
```

```
#include <Rversion.h>
```

Include dependency graph for headers.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define MAXELTSIZE 8192
- #define R_NO_REMAP
- #define STRICT_R_HEADERS
- #define NORET

7.171.1 Macro Definition Documentation

7.171.1.1 MAXELTSIZE

```
#define MAXELTSIZE 8192
```

Definition at line 25 of file headers.h.

7.171.1.2 NORET

```
#define NORET
```

Definition at line 78 of file headers.h.

7.171.1.3 R_NO_REMAP

```
#define R_NO_REMAP
```

Definition at line 26 of file headers.h.

7.171.1.4 STRICT_R_HEADERS

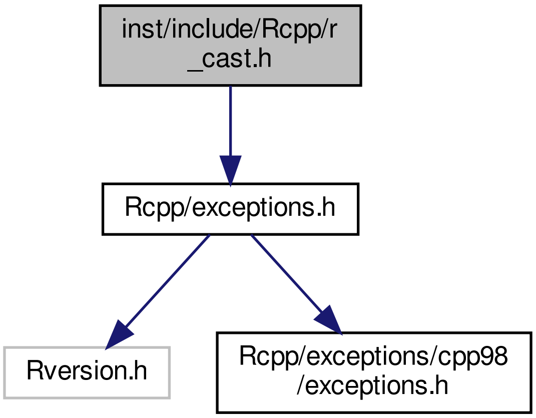
```
#define STRICT_R_HEADERS
```

Definition at line 31 of file headers.h.

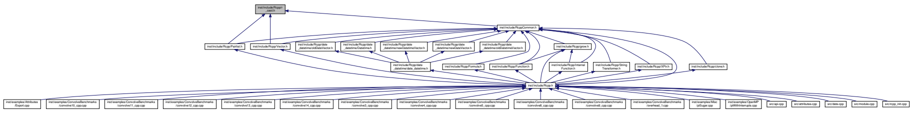
7.172 inst/include/Rcpp/r_cast.h File Reference

```
#include <Rcpp/exceptions.h>
```

Include dependency graph for r_cast.h:


inst/include/Rcpp/r_cast.h
Rcpp/exceptions.h
Rversion.h
Rcpp/exceptions/cpp98/exceptions.h

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

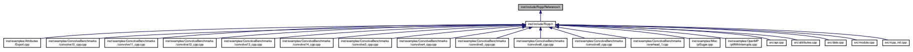
  internal implementation details

Functions

- SEXP Rcpp::internal::convert_using_rfunction (SEXP x, const char ∗const fun)
- template<int TARGET>
  SEXP Rcpp::internal::r_true_cast (SEXP x)
- template<int RTYPE>
  SEXP Rcpp::internal::basic_cast (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< INTSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< REALSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< RAWSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< CPLXSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< LGLSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< STRSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< VECSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< EXPRSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< LISTSXP > (SEXP x)
- template<> SEXP Rcpp::internal::r_true_cast< LANGSXP > (SEXP x)
- template<int TARGET>
  SEXP Rcpp::r_cast (SEXP x)

7.173 inst/include/Rcpp/Reference.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Reference.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef Reference_Impl< PreserveStorage > Rcpp::Reference

Functions

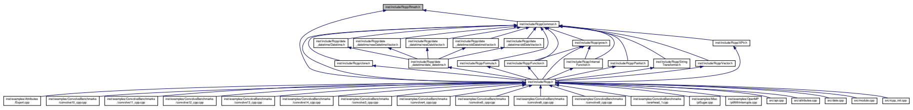
- Rcpp::RCPP_API_CLASS (Reference_Impl)

Variables

- public Rcpp::FieldProxyPolicy< Reference_Impl< StoragePolicy > >

7.174 inst/include/Rcpp/Rmath.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/Rmath.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

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)

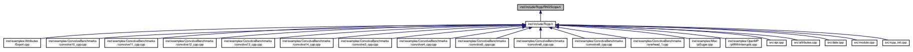
- double R::pbeta (double x, double p, double q, int lt, int lg)
- double R::qbeta (double a, double p, double q, int lt, int lg)
- double R::rbeta (double a, double b)
- double R::dlnorm (double x, double ml, double sl, int lg)
- double R::plnorm (double x, double ml, double sl, int lt, int lg)
- double 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::expm1 (double x)
- double R::log1p (double x)
- int R::imax2 (int x, int y)
- int R::imin2 (int x, int y)
- double R::fmax2 (double x, double y)
- double R::fmin2 (double x, double y)
- double R::sign (double x)
- double R::fprec (double x, double dg)
- double R::fround (double x, double dg)
- double R::fsign (double x, double y)
- double R::ftrunc (double x)

7.175 inst/include/Rcpp/RNGScope.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/RNGScope.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::RNGScope
- class Rcpp::SuspendRNGSynchronizationScope

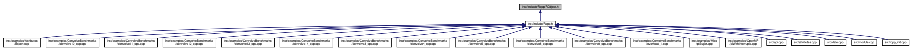
Namespaces

- Rcpp

  Rcpp API.

7.176 inst/include/Rcpp/RObject.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/RObject.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef RObject_Impl< PreserveStorage > Rcpp::RObject

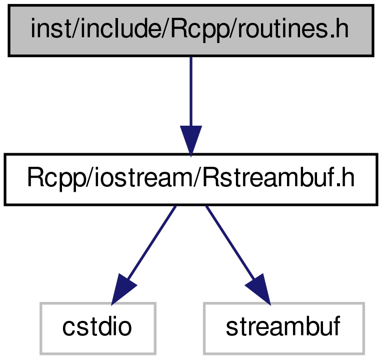
Functions

- Rcpp::RCPP_API_CLASS (RObject_Impl)

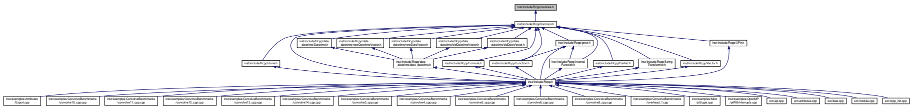
7.177 inst/include/Rcpp/routines.h File Reference

```
#include <Rcpp/iostream/Rstreambuf.h>
```

Include dependency graph for routines.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.
- Rcpp::internal

    internal implementation details

Macros

- #define GET_CALLABLE(__FUN__) (Fun) R_GetCCallable("Rcpp", __FUN__)

Functions

- attribute_hidden const char ∗ Rcpp::type2name (SEXP x)
- attribute_hidden unsigned long Rcpp::internal::enterRNGScope ()
- attribute_hidden unsigned long Rcpp::internal::exitRNGScope ()
- attribute_hidden unsigned long Rcpp::internal::beginSuspendRNGSynchronization ()
- attribute_hidden unsigned long Rcpp::internal::endSuspendRNGSynchronization ()
- attribute_hidden char ∗ Rcpp::internal::get_string_buffer ()
- attribute_hidden SEXP Rcpp::internal::get_Rcpp_namespace ()
- attribute_hidden double Rcpp::mktime00 (struct tm &tm)
- attribute_hidden struct tm ∗ Rcpp::gmtime_ (const time_t ∗const x)
- attribute_hidden void Rcpp::Rcpp_precious_init ()
- attribute_hidden void Rcpp::Rcpp_precious_teardown ()
- attribute_hidden SEXP Rcpp::Rcpp_precious_preserve (SEXP object)
- attribute_hidden void Rcpp::Rcpp_precious_remove (SEXP token)
- attribute_hidden Rostream< true > & Rcpp::Rcpp_cout_get ()
- attribute_hidden Rostream< false > & Rcpp::Rcpp_cerr_get ()
- attribute_hidden SEXP rcpp_get_stack_trace ()
- attribute_hidden SEXP rcpp_set_stack_trace (SEXP e)
- attribute_hidden std::string demangle (const std::string &name)
- attribute_hidden const char ∗ short_file_name (const char ∗file)
- attribute_hidden SEXP stack_trace (const char ∗file="", int line=-1)
- attribute_hidden SEXP get_string_elt (SEXP s, R_xlen_t i)
- attribute_hidden const char ∗ char_get_string_elt (SEXP s, R_xlen_t i)
- attribute_hidden void set_string_elt (SEXP s, R_xlen_t i, SEXP v)
- attribute_hidden void char_set_string_elt (SEXP s, R_xlen_t i, const char ∗v)
- attribute_hidden SEXP ∗ get_string_ptr (SEXP s)
- attribute_hidden SEXP get_vector_elt (SEXP v, R_xlen_t i)
- attribute_hidden void set_vector_elt (SEXP v, R_xlen_t i, SEXP x)
- attribute_hidden SEXP ∗ get_vector_ptr (SEXP v)
- attribute_hidden const char ∗ char_nocheck (SEXP x)
- attribute_hidden void ∗ dataptr (SEXP x)
- attribute_hidden Rcpp::Module ∗ getCurrentScope ()
- attribute_hidden void setCurrentScope (Rcpp::Module ∗mod)
- attribute_hidden int ∗ get_cache (int n)
- attribute_hidden SEXP reset_current_error ()
- attribute_hidden int error_occured ()
- attribute_hidden SEXP rcpp_get_current_error ()

7.177.1 Macro Definition Documentation

7.177.1.1 GET_CALLABLE

```
#define GET_CALLABLE(
             __FUN__ ) (Fun) R_GetCCallable( "Rcpp", __FUN__ )
```

Definition at line 82 of file routines.h.

7.177.2 Function Documentation

7.177.2.1 char_get_string_elt()

```
attribute_hidden const char* char_get_string_elt (
             SEXP s,
             R_xlen_t i ) [inline]
```

Definition at line 216 of file routines.h.

7.177.2.2 char_nocheck()

```
attribute_hidden const char* char_nocheck (
             SEXP x ) [inline]
```

Definition at line 258 of file routines.h.

7.177.2.3 char_set_string_elt()

```
attribute_hidden void char_set_string_elt (
             SEXP s,
             R_xlen_t i,
             const char * v ) [inline]
```

Definition at line 228 of file routines.h.

7.177.2.4 dataptr()

```
attribute_hidden void* dataptr (
             SEXP x )  [inline]
```

Definition at line 264 of file routines.h.

7.177.2.5 demangle()

```
attribute_hidden std::string demangle (
             const std::string & name )  [inline]
```

Definition at line 192 of file routines.h.

7.177.2.6 error_occured()

```
attribute_hidden int error_occured ( )  [inline]
```

Definition at line 294 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.177.2.7 get_cache()

```
attribute_hidden int* get_cache (
             int n )  [inline]
```

Definition at line 282 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and registerFunctions().

7.177.2.8 get_string_elt()

```
attribute_hidden SEXP get_string_elt (
             SEXP s,
             R_xlen_t i )  [inline]
```

Definition at line 210 of file routines.h.

7.177.2.9 get_string_ptr()

```
attribute_hidden SEXP* get_string_ptr (
             SEXP s )  [inline]
```

Definition at line 234 of file routines.h.

7.177.2.10 get_vector_elt()

```
attribute_hidden SEXP get_vector_elt (
             SEXP v,
             R_xlen_t i )  [inline]
```

Definition at line 240 of file routines.h.

7.177.2.11 get_vector_ptr()

```
attribute_hidden SEXP* get_vector_ptr (
             SEXP v )  [inline]
```

Definition at line 252 of file routines.h.

7.177.2.12 getCurrentScope()

```
attribute_hidden Rcpp::Module* getCurrentScope ( )  [inline]
```

Definition at line 270 of file routines.h.

References GET_CALLABLE.

Referenced by class_< Class >::derives(), class_< Class >::get_instance(), registerFunctions(), and Rcpp::enum_< Enum, Parent >::∼enum_().

7.177.2.13 rcpp_get_current_error()

```
attribute_hidden SEXP rcpp_get_current_error ( )  [inline]
```

Definition at line 300 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.177.2.14 rcpp_get_stack_trace()

```
attribute_hidden SEXP rcpp_get_stack_trace ( )  [inline]
```

Definition at line 180 of file routines.h.

References GET_CALLABLE.

Referenced by exception_to_condition_template(), and registerFunctions().

7.177.2.15 rcpp_set_stack_trace()

```
attribute_hidden SEXP rcpp_set_stack_trace (
             SEXP e )  [inline]
```

Definition at line 186 of file routines.h.

References GET_CALLABLE.

Referenced by Rcpp::exception::copy_stack_trace_to_r(), exception_to_condition_template(), and registerFunctions().

7.177.2.16 reset_current_error()

```
attribute_hidden SEXP reset_current_error ( )  [inline]
```

Definition at line 288 of file routines.h.

References GET_CALLABLE.

Referenced by registerFunctions().

7.177.2.17 set_string_elt()

```
attribute_hidden void set_string_elt (
            SEXP s,
            R_xlen_t i,
            SEXP v ) [inline]
```

Definition at line 222 of file routines.h.

7.177.2.18 set_vector_elt()

```
attribute_hidden void set_vector_elt (
            SEXP v,
            R_xlen_t i,
            SEXP x ) [inline]
```

Definition at line 246 of file routines.h.

7.177.2.19 setCurrentScope()

```
attribute_hidden void setCurrentScope (
            Rcpp::Module * mod ) [inline]
```

Definition at line 276 of file routines.h.

References GET_CALLABLE.

Referenced by R_init_Rcpp(), and registerFunctions().

7.177.2.20 short_file_name()

```
attribute_hidden const char* short_file_name (
            const char * file ) [inline]
```

Definition at line 198 of file routines.h.

References GET_CALLABLE.

7.177.2.21 stack_trace()

```
attribute_hidden SEXP stack_trace (
             const char * file = "",
             int line = -1 )  [inline]
```

Definition at line 204 of file routines.h.

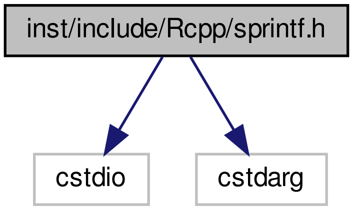
References GET_CALLABLE.

Referenced by registerFunctions().

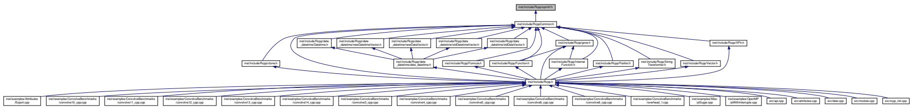
7.178 inst/include/Rcpp/sprintf.h File Reference

```
#include <cstdio>
#include <cstdarg>
```

Include dependency graph for sprintf.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

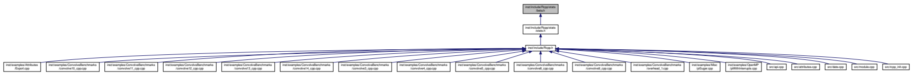
    Rcpp API.

Functions

- template<int MAX_SIZE>
  std::string Rcpp::sprintf (const char ∗format,...)

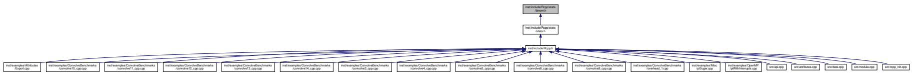
7.179 inst/include/Rcpp/stats/beta.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/beta.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

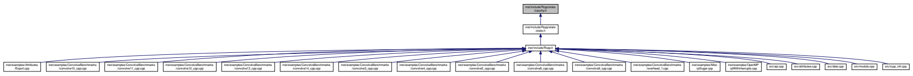
7.180 inst/include/Rcpp/stats/binom.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/binom.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

7.181 inst/include/Rcpp/stats/cauchy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/cauchy.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

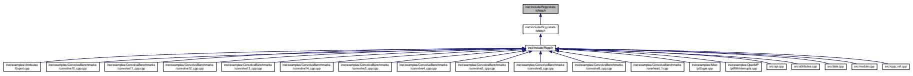
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dcauchy_0 (double x, int give_log)
- double Rcpp::stats::dcauchy_1 (double x, double location, int give_log)
- double Rcpp::stats::pcauchy_0 (double x, int lower_tail, int log_p)
- double Rcpp::stats::pcauchy_1 (double x, double location, int lower_tail, int log_p)
- double Rcpp::stats::qcauchy_0 (double p, int lower_tail, int log_p)
- double Rcpp::stats::qcauchy_1 (double p, double location, int lower_tail, int log_p)

7.182 inst/include/Rcpp/stats/chisq.h File Reference

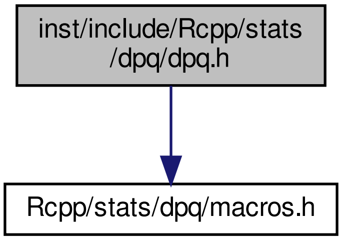
This graph shows which files directly or indirectly include this file:



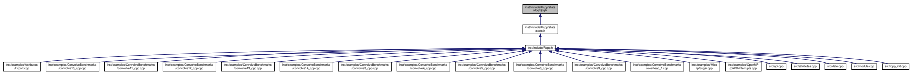
7.183 inst/include/Rcpp/stats/dpq/dpq.h File Reference

`#include <Rcpp/stats/dpq/macros.h>`

Include dependency graph for dpq.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::D0< RTYPE, NA, T >
- class Rcpp::stats::D1< RTYPE, NA, T >
- class Rcpp::stats::D2< RTYPE, NA, T >
- class Rcpp::stats::D3< RTYPE, NA, T >
- class Rcpp::stats::P0< RTYPE, NA, T >
- class Rcpp::stats::P1< RTYPE, NA, T >
- class Rcpp::stats::P2< RTYPE, NA, T >
- class Rcpp::stats::P3< RTYPE, NA, T >
- class Rcpp::stats::Q0< RTYPE, NA, T >
- class Rcpp::stats::Q1< RTYPE, NA, T >
- class Rcpp::stats::Q2< RTYPE, NA, T >
- class Rcpp::stats::Q3< RTYPE, NA, T >

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__)

7.183.1 Macro Definition Documentation

7.183.1.1 RCPP_DPQ_0

```
#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.

7.183.1.2 RCPP_DPQ_1

```
#define RCPP_DPQ_1(
                __NAME__,
                __D__,
                __P__,
                __Q__ )
```

Value:

```
namespace Rcpp {                                                                        \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::D1<RTYPE,NA,T> d##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool log = false                  \
) {                                                                                     \
    return stats::D1<RTYPE,NA,T>( __D__, x, p0, log );                                  \
}                                                                                       \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::P1<RTYPE,NA,T> p##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool lower = true, bool log = false \
) {                                                                                     \
    return stats::P1<RTYPE,NA,T>( __P__, x, p0, lower, log );                           \
}                                                                                       \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::Q1<RTYPE,NA,T> q##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool lower = true, bool log = false \
) {                                                                                     \
    return stats::Q1<RTYPE,NA,T>( __Q__, x, p0, lower, log );                           \
} }
```

Definition at line 340 of file dpq.h.

7.183.1.3 RCPP_DPQ_2

```
#define RCPP_DPQ_2(
                __NAME__,
                __D__,
                __P__,
                __Q__ )
```

Value:

```
namespace Rcpp {                                                                        \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::D2<RTYPE,NA,T> d##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool log = false       \
) {                                                                                     \
    return stats::D2<RTYPE,NA,T>( __D__, x, p0, p1, log );                              \
}                                                                                       \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::P2<RTYPE,NA,T> p##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool lower = true, bool log = false       \
) {                                                                                     \
    return stats::P2<RTYPE,NA,T>( __P__, x, p0, p1, lower, log );                       \
}                                                                                       \
template <int RTYPE, bool NA, typename T>                                               \
inline stats::Q2<RTYPE,NA,T> q##__NAME__(                                               \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool lower = true, bool log = false       \
) {                                                                                     \
    return stats::Q2<RTYPE,NA,T>( __Q__, x, p0, p1, lower, log );                       \
} }
```

Definition at line 363 of file dpq.h.

7.183.1.4 RCPP_DPQ_3

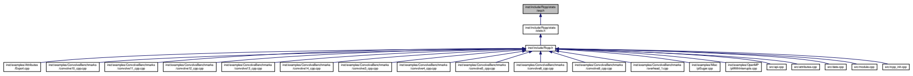
```
#define RCPP_DPQ_3(
                __NAME__,
                __D__,
                __P__,
                __Q__ )
```

Value:
```
namespace Rcpp {                                                                       \
template <int RTYPE, bool NA, typename T>                                              \
inline stats::D3<RTYPE,NA,T> d##__NAME__(                                              \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool log = false
            \
) {                                                                                    \
    return stats::D3<RTYPE,NA,T>( __D__, x, p0, p1, p2, log );                         \
}                                                                                      \
template <int RTYPE, bool NA, typename T>                                              \
inline stats::P3<RTYPE,NA,T> p##__NAME__(                                              \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool lower = true, bool log = false
            \
) {                                                                                    \
    return stats::P3<RTYPE,NA,T>( __P__, x, p0, p1, p2, lower, log );                  \
}                                                                                      \
template <int RTYPE, bool NA, typename T>                                              \
inline stats::Q3<RTYPE,NA,T> q##__NAME__(                                              \
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool lower = true, bool log = false
            \
) {                                                                                    \
    return stats::Q3<RTYPE,NA,T>( __Q__, x, p0, p1, p2, lower, log );                  \
} }
```

Definition at line 386 of file dpq.h.

7.184 inst/include/Rcpp/stats/exp.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

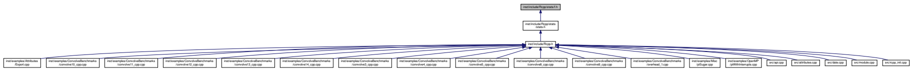
    Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::d_exp_0 (double x, int give_log)
- double Rcpp::stats::q_exp_0 (double p, int lower_tail, int log_p)
- double Rcpp::stats::p_exp_0 (double x, int lower_tail, int log_p)
- template< bool NA, typename T >
  stats::D1< REALSXP, NA, T > Rcpp::dexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool log=false)
- template< bool NA, typename T >
  stats::P1< REALSXP, NA, T > Rcpp::pexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)
- template< bool NA, typename T >
  stats::Q1< REALSXP, NA, T > Rcpp::qexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)

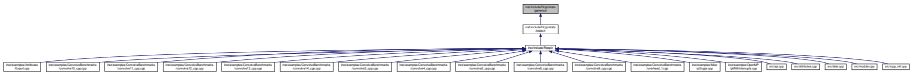
7.185 inst/include/Rcpp/stats/f.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/f.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

7.186 inst/include/Rcpp/stats/gamma.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/gamma.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

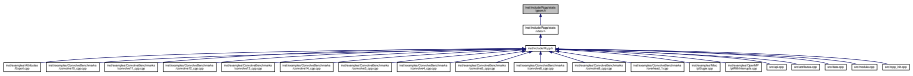
  Rcpp API.
- Rcpp::stats

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)

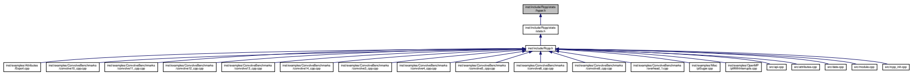
7.187 inst/include/Rcpp/stats/geom.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/geom.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

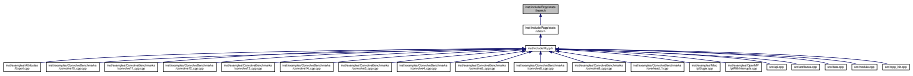
7.188 inst/include/Rcpp/stats/hyper.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/hyper.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

7.189 inst/include/Rcpp/stats/lnorm.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/lnorm.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

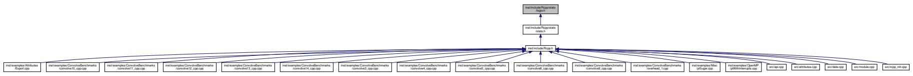
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dlnorm_0 (double x, int log_p)
- double Rcpp::stats::dlnorm_1 (double x, double meanlog, int log_p)
- double Rcpp::stats::plnorm_0 (double x, int lower_tail, int log_p)
- double Rcpp::stats::plnorm_1 (double x, double meanlog, int lower_tail, int log_p)
- double Rcpp::stats::qlnorm_0 (double p, int lower_tail, int log_p)
- double Rcpp::stats::qlnorm_1 (double p, double meanlog, int lower_tail, int log_p)

7.190 inst/include/Rcpp/stats/logis.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/logis.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

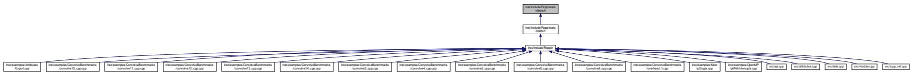
    Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dlogis_0 (double x, int give_log)
- double Rcpp::stats::dlogis_1 (double x, double location, int give_log)
- double Rcpp::stats::plogis_0 (double x, int lower_tail, int log_p)
- double Rcpp::stats::plogis_1 (double x, double location, int lower_tail, int log_p)
- double Rcpp::stats::qlogis_0 (double p, int lower_tail, int log_p)
- double Rcpp::stats::qlogis_1 (double p, double location, int lower_tail, int log_p)

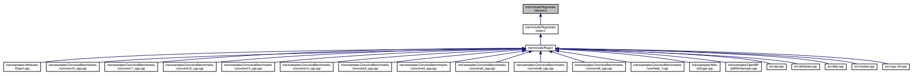
7.191 inst/include/Rcpp/stats/nbeta.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/nbeta.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

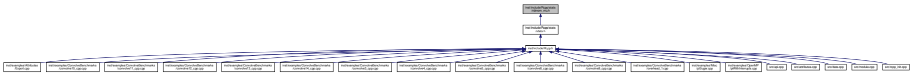
7.192 inst/include/Rcpp/stats/nbinom.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/nbinom.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

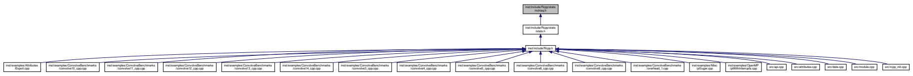
7.193 inst/include/Rcpp/stats/nbinom_mu.h File Reference

This graph shows which files directly or indirectly include this file:



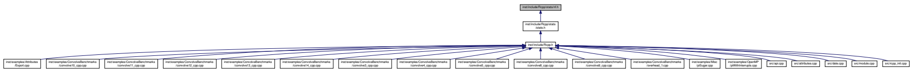
7.194 inst/include/Rcpp/stats/nchisq.h File Reference

This graph shows which files directly or indirectly include this file:



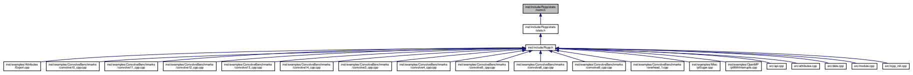
7.195 inst/include/Rcpp/stats/nf.h File Reference

This graph shows which files directly or indirectly include this file:



7.196 inst/include/Rcpp/stats/norm.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

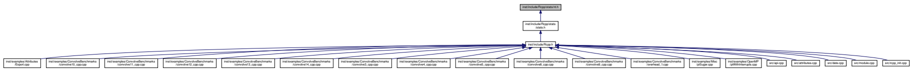
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dnorm_1 (double x, double mu, int give_log)
- double Rcpp::stats::dnorm_0 (double x, int give_log)
- double Rcpp::stats::pnorm_1 (double x, double mu, int lower_tail, int log_p)
- double Rcpp::stats::pnorm_0 (double x, int lower_tail, int log_p)
- double Rcpp::stats::qnorm_1 (double p, double mu, int lower_tail, int log_p)
- double Rcpp::stats::qnorm_0 (double p, int lower_tail, int log_p)

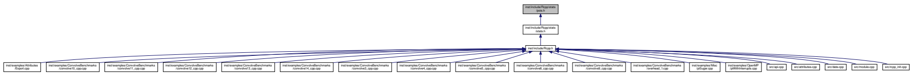
7.197 inst/include/Rcpp/stats/nt.h File Reference

This graph shows which files directly or indirectly include this file:



7.198 inst/include/Rcpp/stats/pois.h File Reference

This graph shows which files directly or indirectly include this file:

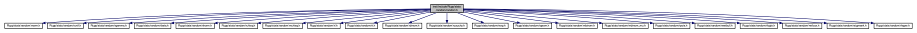


7.199 inst/include/Rcpp/stats/random/random.h File Reference

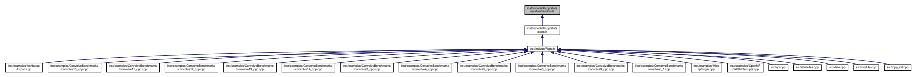
```
#include <Rcpp/stats/random/rnorm.h>
#include <Rcpp/stats/random/runif.h>
#include <Rcpp/stats/random/rgamma.h>
#include <Rcpp/stats/random/rbeta.h>
#include <Rcpp/stats/random/rlnorm.h>
#include <Rcpp/stats/random/rchisq.h>
#include <Rcpp/stats/random/rnchisq.h>
#include <Rcpp/stats/random/rf.h>
#include <Rcpp/stats/random/rt.h>
#include <Rcpp/stats/random/rbinom.h>
#include <Rcpp/stats/random/rcauchy.h>
#include <Rcpp/stats/random/rexp.h>
#include <Rcpp/stats/random/rgeom.h>
#include <Rcpp/stats/random/rnbinom.h>
#include <Rcpp/stats/random/rnbinom_mu.h>
#include <Rcpp/stats/random/rpois.h>
#include <Rcpp/stats/random/rweibull.h>
```

```
#include <Rcpp/stats/random/rlogis.h>
#include <Rcpp/stats/random/rwilcox.h>
#include <Rcpp/stats/random/rsignrank.h>
#include <Rcpp/stats/random/rhyper.h>
```

Include dependency graph for random.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Generator< T >

Namespaces

- Rcpp

  Rcpp API.

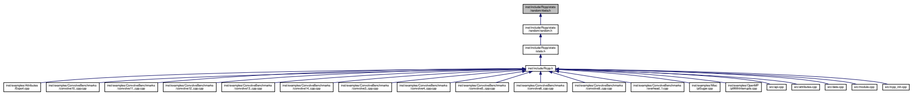
Functions

- NumericVector Rcpp::rnorm (int n, double mean, double sd)
- NumericVector Rcpp::rnorm (int n, double mean)
- NumericVector Rcpp::rnorm (int n)
- NumericVector Rcpp::rbeta (int n, double a, double b)
- NumericVector Rcpp::rbinom (int n, double nin, double pp)
- NumericVector Rcpp::rcauchy (int n, double location, double scale)
- NumericVector Rcpp::rcauchy (int n, double location)
- NumericVector Rcpp::rcauchy (int n)
- NumericVector Rcpp::rchisq (int n, double df)
- NumericVector Rcpp::rexp (int n, double rate)
- NumericVector Rcpp::rexp (int n)
- NumericVector Rcpp::rf (int n, double n1, double n2)
- NumericVector Rcpp::rgamma (int n, double a, double scale)
- NumericVector Rcpp::rgamma (int n, double a)
- NumericVector Rcpp::rgeom (int n, double p)
- NumericVector Rcpp::rhyper (int n, double nn1, double nn2, double kk)
- NumericVector Rcpp::rlnorm (int n, double meanlog, double sdlog)
- NumericVector Rcpp::rlnorm (int n, double meanlog)
- NumericVector Rcpp::rlnorm (int n)
- NumericVector Rcpp::rlogis (int n, double location, double scale)

- NumericVector Rcpp::rlogis (int n, double location)
- NumericVector Rcpp::rlogis (int n)
- NumericVector Rcpp::rnbinom (int n, double siz, double prob)
- NumericVector Rcpp::rnbinom_mu (int n, double siz, double mu)
- NumericVector Rcpp::rnchisq (int n, double df, double lambda)
- NumericVector Rcpp::rnchisq (int n, double df)
- NumericVector Rcpp::rpois (int n, double mu)
- NumericVector Rcpp::rsignrank (int n, double nn)
- NumericVector Rcpp::rt (int n, double df)
- NumericVector Rcpp::runif (int n, double min, double max)
- NumericVector Rcpp::runif (int n, double min)
- NumericVector Rcpp::runif (int n)
- NumericVector Rcpp::rweibull (int n, double shape, double scale)
- NumericVector Rcpp::rweibull (int n, double shape)
- NumericVector Rcpp::rwilcox (int n, double mm, double nn)

7.200 inst/include/Rcpp/stats/random/rbeta.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/random/rbeta.h
inst/include/Rcpp/stats/random/random.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::BetaGenerator

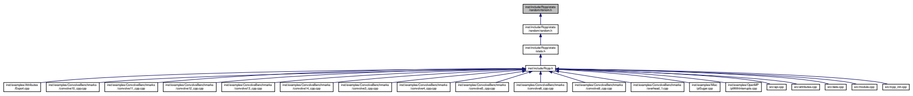
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.201 inst/include/Rcpp/stats/random/rbinom.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/random/rbinom.h
inst/include/Rcpp/stats/random/random.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::BinomGenerator

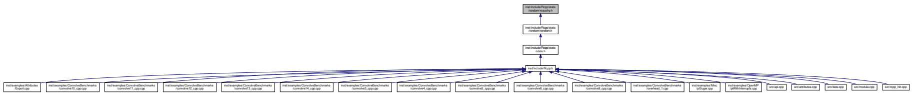
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.202 inst/include/Rcpp/stats/random/rcauchy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/random/rcauchy.h
inst/include/Rcpp/stats/random/random.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::CauchyGenerator
- class Rcpp::stats::CauchyGenerator_1
- class Rcpp::stats::CauchyGenerator_0

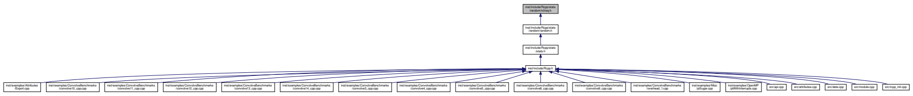
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.203 inst/include/Rcpp/stats/random/rchisq.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/random/rchisq.h
inst/include/Rcpp/stats/random/random.h
inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::ChisqGenerator

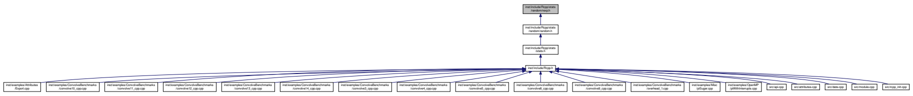
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.204 inst/include/Rcpp/stats/random/rexp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::ExpGenerator
- class Rcpp::stats::ExpGenerator__rate1

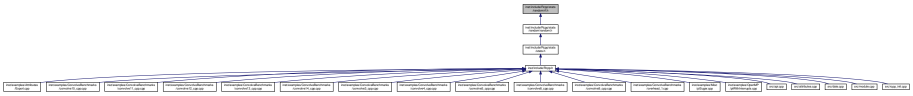
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.205 inst/include/Rcpp/stats/random/rf.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::FGenerator_Finite_Finite
- class Rcpp::stats::FGenerator_NotFinite_Finite
- class Rcpp::stats::FGenerator_Finite_NotFinite

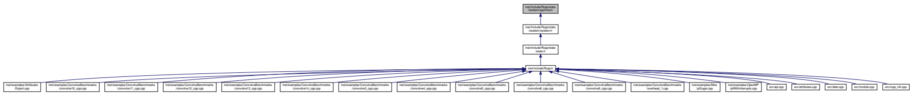
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.206 inst/include/Rcpp/stats/random/rgamma.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::GammaGenerator

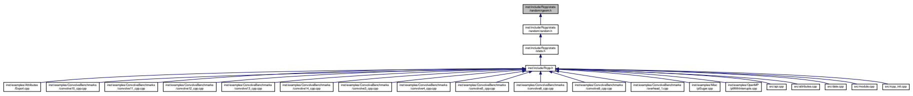
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.207 inst/include/Rcpp/stats/random/rgeom.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::GeomGenerator

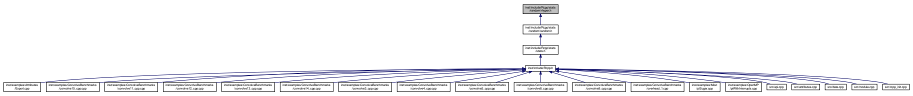
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.208 inst/include/Rcpp/stats/random/rhyper.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::HyperGenerator

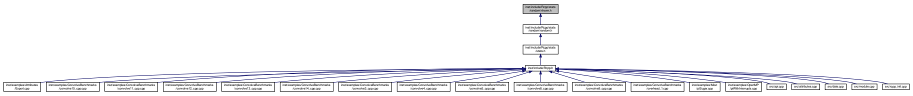
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.209 inst/include/Rcpp/stats/random/rlnorm.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::LNormGenerator
- class Rcpp::stats::LNormGenerator_1
- class Rcpp::stats::LNormGenerator_0

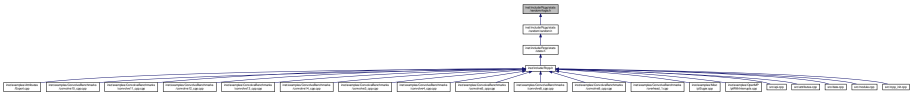
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.210 inst/include/Rcpp/stats/random/rlogis.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::LogisGenerator
- class Rcpp::stats::LogisGenerator_1
- class Rcpp::stats::LogisGenerator_0

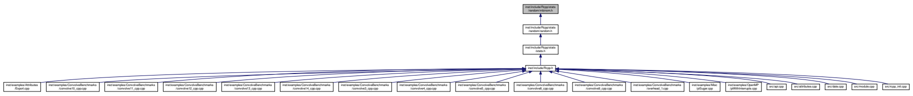
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.211 inst/include/Rcpp/stats/random/rnbinom.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::NBinomGenerator

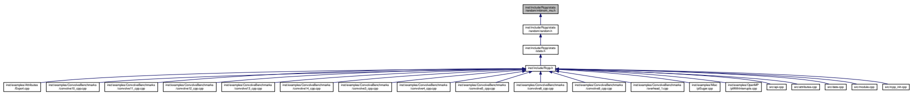
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.212 inst/include/Rcpp/stats/random/rnbinom_mu.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::NBinomGenerator_Mu

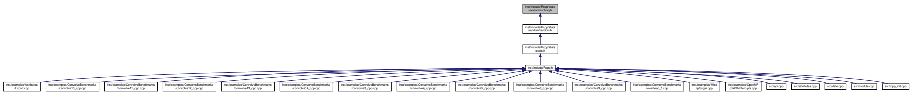
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.213 inst/include/Rcpp/stats/random/rnchisq.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::NChisqGenerator

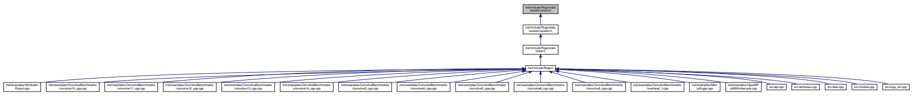
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.214 inst/include/Rcpp/stats/random/rnorm.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::NormGenerator
- class Rcpp::stats::NormGenerator__sd1
- class Rcpp::stats::NormGenerator__mean0
- class Rcpp::stats::NormGenerator__mean0__sd1

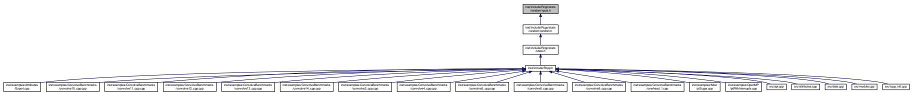
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.215 inst/include/Rcpp/stats/random/rpois.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::PoissonGenerator

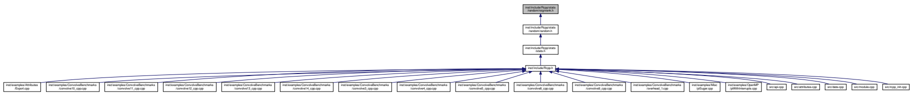
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.216 inst/include/Rcpp/stats/random/rsignrank.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats /random/rsignrank.h
inst/include/Rcpp/stats /random/random.h
inst/include/Rcpp/stats /stats.h
inst/include/Rcpp.h
inst/examples/Attributes /Export.cpp
inst/examples/ConvolveBenchmarks /convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks /overhead_1.cpp
inst/examples/Misc /piSugar.cpp
inst/examples/OpenMP /piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::SignRankGenerator

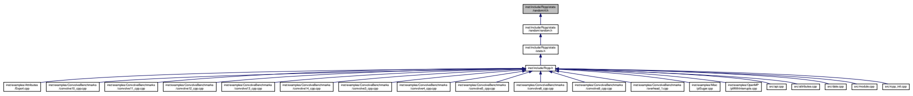
Namespaces

- Rcpp

    Rcpp API.
- Rcpp::stats

7.217 inst/include/Rcpp/stats/random/rt.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats /random/rt.h
inst/include/Rcpp/stats /random/random.h
inst/include/Rcpp/stats /stats.h
inst/include/Rcpp.h
inst/examples/Attributes /Export.cpp
inst/examples/ConvolveBenchmarks /convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks /overhead_1.cpp
inst/examples/Misc /piSugar.cpp
inst/examples/OpenMP /piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::stats::TGenerator

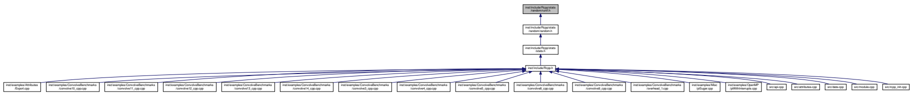
Namespaces

- Rcpp

    Rcpp API.
- Rcpp::stats

7.218 inst/include/Rcpp/stats/random/runif.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::UnifGenerator
- class Rcpp::stats::UnifGenerator__0__1

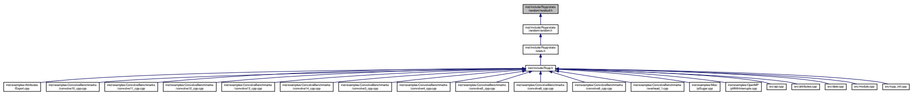
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.219 inst/include/Rcpp/stats/random/rweibull.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::WeibullGenerator
- class Rcpp::stats::WeibullGenerator__scale1

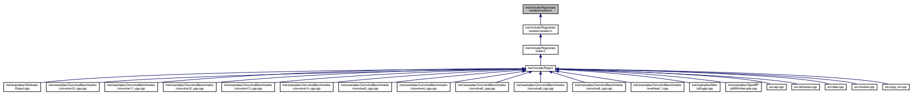
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::stats

7.220 inst/include/Rcpp/stats/random/rwilcox.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::stats::WilcoxGenerator

Namespaces

- Rcpp

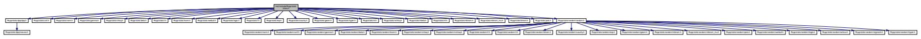
  Rcpp API.
- Rcpp::stats

7.221 inst/include/Rcpp/stats/stats.h File Reference

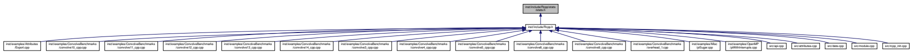
```
#include <Rcpp/stats/dpq/dpq.h>
#include <Rcpp/stats/unif.h>
#include <Rcpp/stats/norm.h>
#include <Rcpp/stats/gamma.h>
#include <Rcpp/stats/chisq.h>
#include <Rcpp/stats/beta.h>
#include <Rcpp/stats/t.h>
#include <Rcpp/stats/lnorm.h>
#include <Rcpp/stats/weibull.h>
#include <Rcpp/stats/logis.h>
#include <Rcpp/stats/f.h>
#include <Rcpp/stats/exp.h>
#include <Rcpp/stats/cauchy.h>
#include <Rcpp/stats/geom.h>
#include <Rcpp/stats/hyper.h>
#include <Rcpp/stats/nt.h>
#include <Rcpp/stats/nchisq.h>
#include <Rcpp/stats/nbeta.h>
#include <Rcpp/stats/nf.h>
#include <Rcpp/stats/nbinom.h>
#include <Rcpp/stats/nbinom_mu.h>
#include <Rcpp/stats/binom.h>
#include <Rcpp/stats/pois.h>
```

```
#include <Rcpp/stats/random/random.h>
```

Include dependency graph for stats.h:



This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/stats/stats.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Macros

- #define ML_POSINF R_PosInf
- #define ML_NEGINF R_NegInf
- #define ML_NAN R_NaN

7.221.1 Macro Definition Documentation

7.221.1.1 ML_NAN

```
#define ML_NAN R_NaN
```

Definition at line 29 of file stats.h.

7.221.1.2 ML_NEGINF

```
#define ML_NEGINF R_NegInf
```

Definition at line 28 of file stats.h.

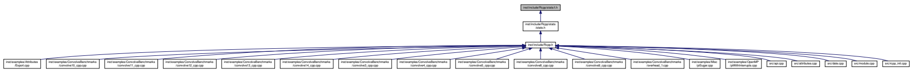
7.221.1.3 ML_POSINF

```
#define ML_POSINF R_PosInf
```

Definition at line 27 of file stats.h.

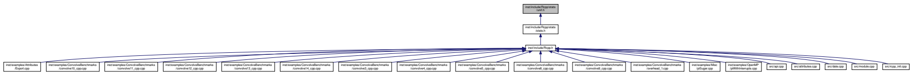
7.222 inst/include/Rcpp/stats/t.h File Reference

This graph shows which files directly or indirectly include this file:



7.223 inst/include/Rcpp/stats/unif.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

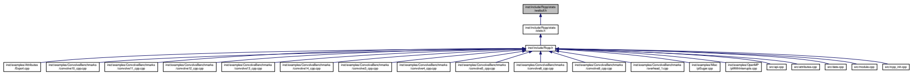
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dunif_1 (double x, double a, int give_log)
- double Rcpp::stats::dunif_0 (double x, int give_log)
- double Rcpp::stats::punif_1 (double x, double a, int lower_tail, int log_p)
- double Rcpp::stats::punif_0 (double x, int lower_tail, int log_p)
- double Rcpp::stats::qunif_1 (double p, double a, int lower_tail, int log_p)
- double Rcpp::stats::qunif_0 (double p, int lower_tail, int log_p)

7.224 inst/include/Rcpp/stats/weibull.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

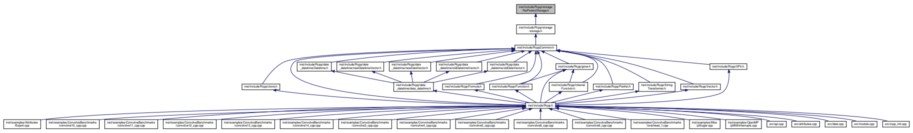
  Rcpp API.
- Rcpp::stats

Functions

- double Rcpp::stats::dweibull_1 (double x, double shape, int give_log)
- double Rcpp::stats::pweibull_1 (double x, double shape, int lower_tail, int log_p)
- double Rcpp::stats::qweibull_1 (double p, double shape, int lower_tail, int log_p)

7.225 inst/include/Rcpp/storage/NoProtectStorage.h File Reference

This graph shows which files directly or indirectly include this file:



inst/include/Rcpp/storage /NoProtectStorage.h
inst/include/Rcpp/storage /storage.h
inst/include/RcppCommon.h
inst/include/Rcpp/date _datetime/Datetime.h
inst/include/Rcpp/date _datetime/newDatetimeVector.h
inst/include/Rcpp/date _datetime/newDateVector.h
inst/include/Rcpp/date _datetime/oldDatetimeVector.h
inst/include/Rcpp/date _datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date _datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes /Export.cpp
inst/examples/ConvolveBenchmarks /convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks /overhead_1.cpp
inst/examples/Misc /piSugar.cpp
inst/examples/OpenMP /piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::NoProtectStorage< CLASS >

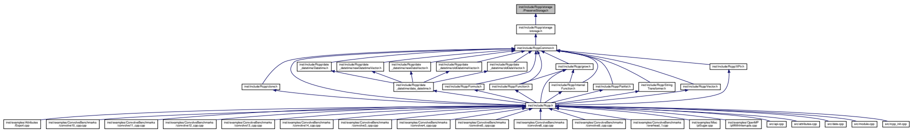
Namespaces

- Rcpp

  Rcpp API.

7.226 inst/include/Rcpp/storage/PreserveStorage.h File Reference

This graph shows which files directly or indirectly include this file:



inst/include/Rcpp/storage /PreserveStorage.h
inst/include/Rcpp/storage /storage.h
inst/include/RcppCommon.h
inst/include/Rcpp/date _datetime/Datetime.h
inst/include/Rcpp/date _datetime/newDatetimeVector.h
inst/include/Rcpp/date _datetime/newDateVector.h
inst/include/Rcpp/date _datetime/oldDatetimeVector.h
inst/include/Rcpp/date _datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date _datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes /Export.cpp
inst/examples/ConvolveBenchmarks /convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks /convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks /overhead_1.cpp
inst/examples/Misc /piSugar.cpp
inst/examples/OpenMP /piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::PreserveStorage< CLASS >

Namespaces

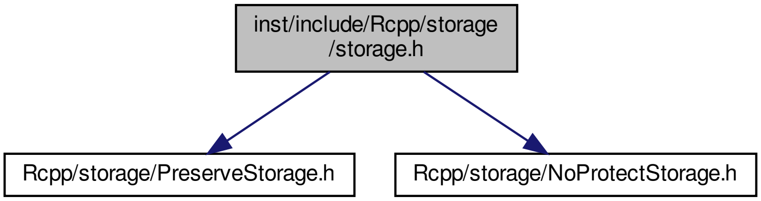
- Rcpp

  Rcpp API.

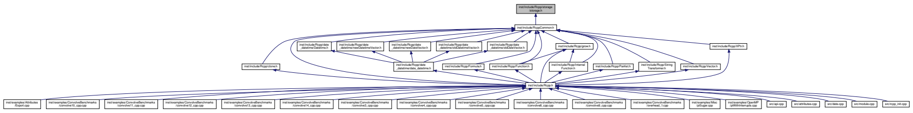
7.227 inst/include/Rcpp/storage/storage.h File Reference

```
#include <Rcpp/storage/PreserveStorage.h>
#include <Rcpp/storage/NoProtectStorage.h>
```

Include dependency graph for storage.h:

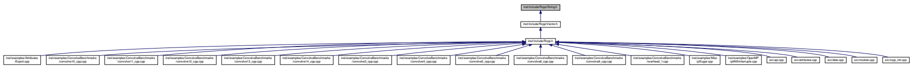

inst/include/Rcpp/storage
/storage.h
Rcpp/storage/PreserveStorage.h
Rcpp/storage/NoProtectStorage.h

This graph shows which files directly or indirectly include this file:



7.228 inst/include/Rcpp/String.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::String
- struct Rcpp::traits::r_type_traits< Rcpp::String >
- struct Rcpp::traits::r_sexptype_traits< Rcpp::String >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap
- Rcpp::internal

  internal implementation details

Macros

- #define RCPP_STRING_DEBUG_LEVEL 0
- #define RCPP_STRING_DEBUG(MSG)
- #define RCPP_STRING_DEBUG_1(fmt, MSG)
- #define RCPP_STRING_DEBUG_2(fmt, M1, M2)
- #define RCPP_STRING_DEBUG_3(fmt, M1, M2, M3)

Functions

- template<> SEXP Rcpp::internal::make_charsexp< Rcpp::String > (const Rcpp::String &s)
- template<> SEXP Rcpp::wrap< Rcpp::String > (const Rcpp::String &object)
- bool Rcpp::operator== (const String::StringProxy &lhs, const String &rhs)
- bool Rcpp::operator!= (const String::StringProxy &lhs, const String &rhs)
- bool Rcpp::operator== (const String::const_StringProxy &lhs, const String &rhs)
- bool Rcpp::operator!= (const String::const_StringProxy &lhs, const String &rhs)

7.228.1 Macro Definition Documentation

7.228.1.1 RCPP_STRING_DEBUG

```
#define RCPP_STRING_DEBUG(
              MSG )
```

Definition at line 37 of file String.h.

7.228.1.2 RCPP_STRING_DEBUG_1

```
#define RCPP_STRING_DEBUG_1(
              fmt,
              MSG )
```

Definition at line 38 of file String.h.

7.228.1.3 RCPP_STRING_DEBUG_2

```
#define RCPP_STRING_DEBUG_2(
              fmt,
              M1,
              M2 )
```

Definition at line 39 of file String.h.

7.228.1.4 RCPP_STRING_DEBUG_3

```
#define RCPP_STRING_DEBUG_3(
              fmt,
              M1,
              M2,
              M3 )
```

Definition at line 40 of file String.h.

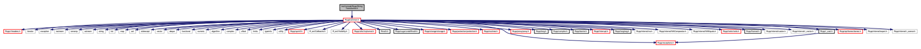
7.228.1.5 RCPP_STRING_DEBUG_LEVEL

```
#define RCPP_STRING_DEBUG_LEVEL 0
```

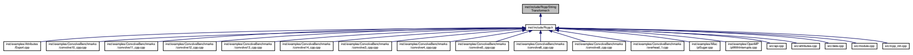
Definition at line 26 of file String.h.

7.229 inst/include/Rcpp/StringTransformer.h File Reference

`#include <RcppCommon.h>`
Include dependency graph for StringTransformer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::StringTransformer< UnaryOperator >

Namespaces

- Rcpp

  Rcpp API.

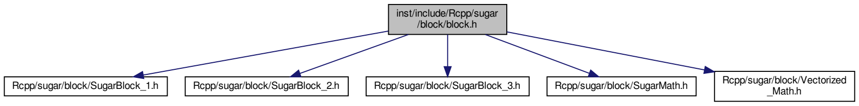
Functions

- template<typename UnaryOperator >
  StringTransformer< UnaryOperator > Rcpp::make_string_transformer (const UnaryOperator &fun)

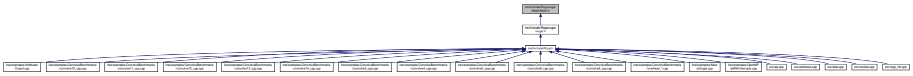
7.230 inst/include/Rcpp/sugar/block/block.h File Reference

```
#include <Rcpp/sugar/block/SugarBlock_1.h>
#include <Rcpp/sugar/block/SugarBlock_2.h>
#include <Rcpp/sugar/block/SugarBlock_3.h>
#include <Rcpp/sugar/block/SugarMath.h>
#include <Rcpp/sugar/block/Vectorized_Math.h>
```

Include dependency graph for block.h:

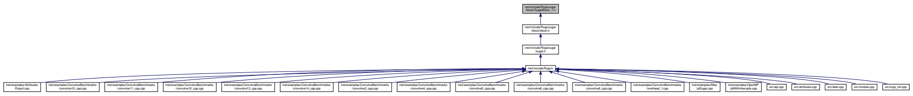

inst/include/Rcpp/sugar
/block/block.h
Rcpp/sugar/block/SugarBlock_1.h
Rcpp/sugar/block/SugarBlock_2.h
Rcpp/sugar/block/SugarBlock_3.h
Rcpp/sugar/block/SugarMath.h
Rcpp/sugar/block/Vectorized
_Math.h

This graph shows which files directly or indirectly include this file:



7.231 inst/include/Rcpp/sugar/block/SugarBlock_1.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define SB1_T VectorBase<REALSXP,NA,T>
- #define SUGAR_BLOCK_1(__NAME__, __SYMBOL__)

7.231.1 Macro Definition Documentation

7.231.1.1 SB1_T

```
#define SB1_T VectorBase<REALSXP,NA,T>
```

Definition at line 47 of file SugarBlock_1.h.

7.231.1.2 SUGAR_BLOCK_1

```
#define SUGAR_BLOCK_1(
                __NAME__,
                __SYMBOL__ )
```

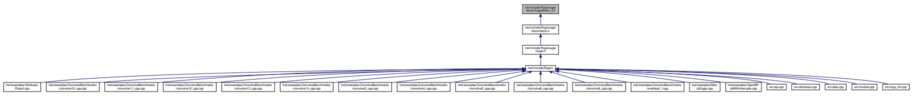
Value:

```
    namespace Rcpp{                                                          \
    template <bool NA, typename T>                                           \
    inline sugar::SugarBlock_1<NA,double,double,SB1_T >                      \
    __NAME__(                                                                \
        const SB1_T& t                                                       \
    ){                                                                       \
        return sugar::SugarBlock_1<NA,double,double,SB1_T >(                 \
                __SYMBOL__ , t                                               \
        ) ;                                                                  \
    }                                                                        \
}
```

Definition at line 49 of file SugarBlock_1.h.

7.232 inst/include/Rcpp/sugar/block/SugarBlock_2.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >
- class Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >
- class Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define SB2_LHT VectorBase<REALSXP,LHS_NA,LHS_T>
- #define SB2_RHT VectorBase<REALSXP,RHS_NA,RHS_T>
- #define SUGAR_BLOCK_2(__NAME__, __SYMBOL__)
- #define SUGAR_BLOCK_2_NA(__NAME__, __SYMBOL__, __NA__)

7.232.1 Macro Definition Documentation

7.232.1.1 SB2_LHT

```
#define SB2_LHT VectorBase<REALSXP,LHS_NA,LHS_T>
```

Definition at line 89 of file SugarBlock_2.h.

7.232.1.2 SB2_RHT

```
#define SB2_RHT VectorBase<REALSXP,RHS_NA,RHS_T>
```

Definition at line 90 of file SugarBlock_2.h.

7.232.1.3 SUGAR_BLOCK_2

```
#define SUGAR_BLOCK_2(
              __NAME__,
              __SYMBOL__ )
```

Definition at line 92 of file SugarBlock_2.h.

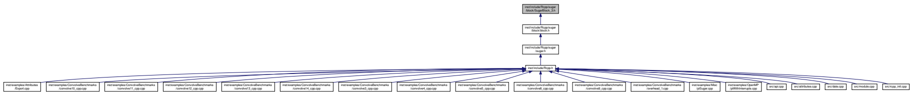
7.232.1.4 SUGAR_BLOCK_2_NA

```
#define SUGAR_BLOCK_2_NA(
              __NAME__,
              __SYMBOL__,
              __NA__ )
```

Definition at line 127 of file SugarBlock_2.h.

7.233 inst/include/Rcpp/sugar/block/SugarBlock_3.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/block/SugarBlock_3.h
inst/include/Rcpp/sugar/block/block.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define SB3_T1 VectorBase<REALSXP,T1_NA,T1>
- #define SB3_T2 VectorBase<REALSXP,T2_NA,T2>
- #define SB3_T3 VectorBase<REALSXP,T3_NA,T3>
- #define SUGAR_BLOCK_3(__NAME__, __SYMBOL__)

7.233.1 Macro Definition Documentation

7.233.1.1 SB3_T1

```
#define SB3_T1 VectorBase<REALSXP,T1_NA,T1>
```

Definition at line 97 of file SugarBlock_3.h.

7.233.1.2 SB3_T2

```
#define SB3_T2 VectorBase<REALSXP,T2_NA,T2>
```

Definition at line 98 of file SugarBlock_3.h.

7.233.1.3 SB3_T3

```
#define SB3_T3 VectorBase<REALSXP,T3_NA,T3>
```

Definition at line 99 of file SugarBlock_3.h.

7.233.1.4 SUGAR_BLOCK_3

```
#define SUGAR_BLOCK_3(
                __NAME__,
                __SYMBOL__ )
```

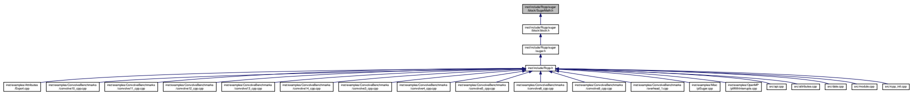
Value:

```
namespace Rcpp{                                                                              \
        template <bool T1_NA, typename T1, bool T2_NA, typename T2, bool T3_NA, typename T3>  \
        inline sugar::SugarBlock_3_VVV<                                                       \
                (T1_NA||T2_NA||T3_NA) ,double,                                                \
                double,SB3_T1,                                                                \
                double,SB3_T2,                                                                \
                double,SB3_T3                                                                 \
        >                                                                                     \
        __NAME__(                                                                             \
                const SB3_T1& x1,                                                             \
                const SB3_T2& x2,                                                             \
                const SB3_T3& x3                                                              \
        ){                                                                                    \
                return sugar::SugarBlock_3_VVV<                                               \
                        (T1_NA||T2_NA||T3_NA) , double,                                       \
                        double,SB3_T1,                                                        \
                        double,SB3_T2,                                                        \
                        double,SB3_T3                                                         \
                >(                                                                            \
                        __SYMBOL__ , x1, x2, x3                                               \
                ) ;                                                                           \
        }                                                                                     \
}
```

Definition at line 101 of file SugarBlock_3.h.

7.234 inst/include/Rcpp/sugar/block/SugarMath.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >
- class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >
- class Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define SUGAR_MATH_1(__NAME__, __SYMBOL__)

7.234.1 Macro Definition Documentation

7.234.1.1 SUGAR_MATH_1

```
#define SUGAR_MATH_1(
                __NAME__,
                __SYMBOL__ )
```

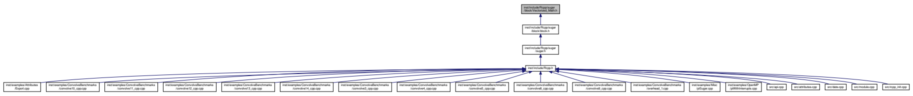
Value:

```
        namespace Rcpp{                                                             \
        template <bool NA, typename T>                                              \
        inline sugar::SugarMath_1<NA,double,double,T, double (*)(double) >          \
        __NAME__(                                                                   \
                const VectorBase<REALSXP,NA,T>& t                                       \
        ){                                                                          \
                return sugar::SugarMath_1<NA,double,double,T, double (*)(double)>(      \
                        &__SYMBOL__ , t                                                      \
                ) ;                                                                     \
        }                                                                           \
        inline sugar::SugarMath_1<true,double,double,NumericVector,double(*)(double)>   \
        __NAME__( SEXP x){ return __NAME__( NumericVector( x ) ) ; }                \
        template <bool NA, typename T>                                              \
        inline sugar::SugarMath_1<NA,double,int,T, double (*)(double) >             \
        __NAME__(                                                                   \
                const VectorBase<INTSXP,NA,T>& t                                        \
        ){                                                                          \
                return sugar::SugarMath_1<NA,double,int,T, double (*)(double)>(         \
                        &__SYMBOL__ , t                                                      \
                ) ;                                                                     \
        }                                                                           \
        }
```

Definition at line 100 of file SugarMath.h.

7.235 inst/include/Rcpp/sugar/block/Vectorized_Math.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/block/Vectorized_Math.h
inst/include/Rcpp/sugar/block/block.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Vectorized< Func, NA, VEC >
- class Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >
- class Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define VECTORIZED_MATH_1(__NAME__, __SYMBOL__)

Typedefs

- typedef double(∗ Rcpp::sugar::DDFun) (double)

7.235.1 Macro Definition Documentation

7.235.1.1 VECTORIZED_MATH_1

```
#define VECTORIZED_MATH_1(
                __NAME__,
                __SYMBOL__ )
```

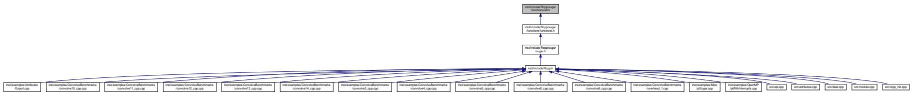
Value:

```
namespace Rcpp{                                                             \
        template <bool NA, typename T>                                          \
        inline sugar::Vectorized<__SYMBOL__,NA,T>                               \
        __NAME__( const VectorBase<REALSXP,NA,T>& t ){                          \
                return sugar::Vectorized<__SYMBOL__,NA,T>( t ) ;                    \
        }                                                                       \
        inline sugar::Vectorized<__SYMBOL__,true,NumericVector>                 \
        __NAME__( SEXP x){ return __NAME__( NumericVector( x ) ) ; }            \
        template <bool NA, typename T>                                          \
        inline sugar::Vectorized_INTSXP<__SYMBOL__,NA,T>                        \
        __NAME__( const VectorBase<INTSXP,NA,T>& t      ){                          \
                return sugar::Vectorized_INTSXP<__SYMBOL__,NA,T>( t ) ;             \
        }                                                                       \
}
```

Definition at line 83 of file Vectorized_Math.h.

7.236 inst/include/Rcpp/sugar/functions/all.h File Reference

This graph shows which files directly or indirectly include this file:



inst/include/Rcpp/sugar/functions/all.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::All< NA, T >
- class Rcpp::sugar::All< false, T >

Namespaces

- Rcpp

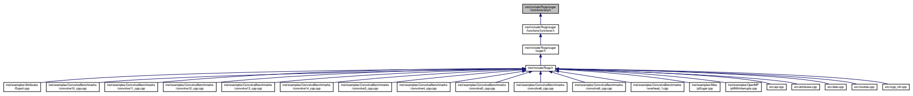
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::All< NA, T > Rcpp::all (const Rcpp::VectorBase< LGLSXP, NA, T > &t)

7.237 inst/include/Rcpp/sugar/functions/any.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/any.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Any< NA, T >
- class Rcpp::sugar::Any< false, T >

Namespaces

- Rcpp

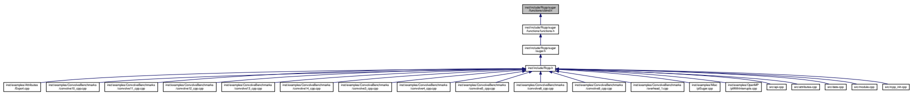
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Any< NA, T > Rcpp::any (const Rcpp::VectorBase< LGLSXP, NA, T > &t)

7.238 inst/include/Rcpp/sugar/functions/cbind.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/cbind.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >
- struct Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP >
- struct Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >
- struct Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP >
- class Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >
- class Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >
- struct Rcpp::sugar::cbind_impl::scalar< RTYPE >
- class Rcpp::sugar::cbind_impl::ScalarBindable< T >
- class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >
- class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >
- class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >
- class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typena
- class Rcpp::sugar::cbind_impl::detail::has_stored_type< T >
- struct Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no
- struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container >
- struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >
- struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >
- struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >
- struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false >
- struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >
- struct Rcpp::sugar::cbind_impl::matrix_return< T, B >
- struct Rcpp::sugar::cbind_impl::matrix_return< T, false >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar
- Rcpp::sugar::cbind_impl
- Rcpp::sugar::cbind_impl::detail

Macros

- #define MakeBindable(x) (cbind_impl::detail::MakeBindable(x)(x))

Functions

- template<int RTYPE>
  ContainerBindable< RTYPE, Rcpp::Matrix< RTYPE > > Rcpp::sugar::cbind_impl::MakeContainerBindable (const Rcpp::Matrix< RTYPE > &x)
- template<int RTYPE>
  ContainerBindable< RTYPE, Rcpp::Vector< RTYPE > > Rcpp::sugar::cbind_impl::MakeContainerBindable (const Rcpp::Vector< RTYPE > &x)
- template<> ContainerBindable< LGLSXP, Rcpp::Matrix< LGLSXP > > Rcpp::sugar::cbind_impl::MakeContainerBindable (const Rcpp::Matrix< LGLSXP > &x)
- template<> ContainerBindable< LGLSXP, Rcpp::Vector< LGLSXP > > Rcpp::sugar::cbind_impl::MakeContainerBindable (const Rcpp::Vector< LGLSXP > &x)
- template<typename T >
  ScalarBindable< T > Rcpp::sugar::cbind_impl::MakeScalarBindable (const T &t)
- template<int RTYPE, typename E1 , typename E2 >
  JoinOp< RTYPE, E1, E2 > Rcpp::sugar::cbind_impl::operator, (const BindableExpression< RTYPE, E1 > &e1, const BindableExpression< RTYPE, E2 > &e2)
- template<typename T >
  Rcpp::traits::enable_if< has_stored_type< T >::value, MakeBindableCall< T, true > >::type Rcpp::sugar::cbind_impl::detail::Make
  (const T &t)
- template<typename T >
  Rcpp::traits::enable_if< !has_stored_type< T >::value, MakeBindableCall< T, false > >::type Rcpp::sugar::cbind_impl::detail::Mak
  (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)

7.238.1 Macro Definition Documentation

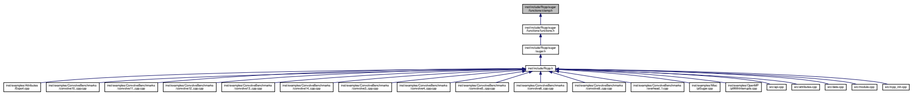
7.238.1.1 MakeBindable

```
#define MakeBindable(
              x ) (cbind_impl::detail::MakeBindable(x)(x))
```

Definition at line 510 of file cbind.h.

7.239 inst/include/Rcpp/sugar/functions/clamp.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/clamp.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::sugar::clamp_operator< RTYPE, NA >
- struct Rcpp::sugar::clamp_operator< REALSXP, true >
- class Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >

Namespaces

- Rcpp

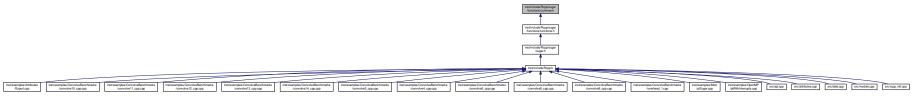
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > Rcpp::clamp (typename Rcpp::traits::storage_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, NA, T > &vec, typename Rcpp::traits::storage_type< RTYPE >::type rhs)

7.240 inst/include/Rcpp/sugar/functions/cummax.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Cummax< RTYPE, NA, T >

Namespaces

- Rcpp

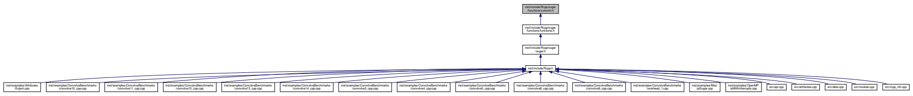
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Cummax< INTSXP, NA, T > Rcpp::cummax (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cummax< REALSXP, NA, T > Rcpp::cummax (const VectorBase< REALSXP, NA, T > &t)

7.241 inst/include/Rcpp/sugar/functions/cummin.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Cummin< RTYPE, NA, T >

Namespaces

- Rcpp

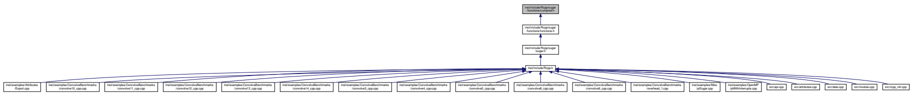
  Rcpp API.
- Rcpp::sugar

Functions

- template< bool NA, typename T >
  sugar::Cummin< INTSXP, NA, T > Rcpp::cummin (const VectorBase< INTSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Cummin< REALSXP, NA, T > Rcpp::cummin (const VectorBase< REALSXP, NA, T > &t)

7.242 inst/include/Rcpp/sugar/functions/cumprod.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Cumprod< RTYPE, NA, T >

Namespaces

- Rcpp

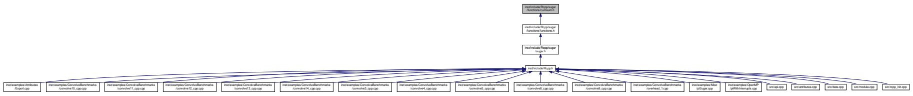
  Rcpp API.
- Rcpp::sugar

Functions

- template< bool NA, typename T >
  sugar::Cumprod< INTSXP, NA, T > Rcpp::cumprod (const VectorBase< INTSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Cumprod< REALSXP, NA, T > Rcpp::cumprod (const VectorBase< REALSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Cumprod< CPLXSXP, NA, T > Rcpp::cumprod (const VectorBase< CPLXSXP, NA, T > &t)

7.243 inst/include/Rcpp/sugar/functions/cumsum.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Cumsum< RTYPE, NA, T >

Namespaces

- Rcpp

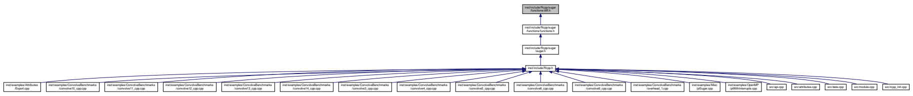
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Cumsum< INTSXP, NA, T > Rcpp::cumsum (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Cumsum< REALSXP, NA, T > Rcpp::cumsum (const VectorBase< REALSXP, NA, T > &t)

7.244 inst/include/Rcpp/sugar/functions/diff.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/diff.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >
- class Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >
- class Rcpp::sugar::Diff< RTYPE, false, LHS_T >

Namespaces

- Rcpp

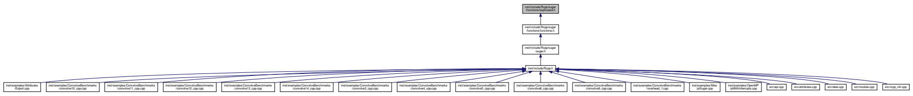
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool LHS_NA, typename LHS_T >
  sugar::Diff< INTSXP, LHS_NA, LHS_T > Rcpp::diff (const VectorBase< INTSXP, LHS_NA, LHS_T > &lhs)
- template<bool LHS_NA, typename LHS_T >
  sugar::Diff< REALSXP, LHS_NA, LHS_T > Rcpp::diff (const VectorBase< REALSXP, LHS_NA, LHS_T > &lhs)

7.245 inst/include/Rcpp/sugar/functions/duplicated.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Functions

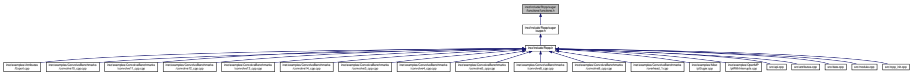
- template< int RTYPE, bool NA, typename T >
  LogicalVector Rcpp::duplicated (const VectorBase< RTYPE, NA, T > &x)

7.246 inst/include/Rcpp/sugar/functions/functions.h File Reference

```
#include <cmath>
#include <Rcpp/sugar/functions/Lazy.h>
#include <Rcpp/sugar/functions/math.h>
#include <Rcpp/sugar/functions/complex.h>
#include <Rcpp/sugar/functions/any.h>
#include <Rcpp/sugar/functions/all.h>
#include <Rcpp/sugar/functions/is_na.h>
#include <Rcpp/sugar/functions/is_finite.h>
#include <Rcpp/sugar/functions/is_infinite.h>
#include <Rcpp/sugar/functions/is_nan.h>
#include <Rcpp/sugar/functions/na_omit.h>
#include <Rcpp/sugar/functions/seq_along.h>
#include <Rcpp/sugar/functions/sapply.h>
#include <Rcpp/sugar/functions/mapply.h>
#include <Rcpp/sugar/functions/lapply.h>
#include <Rcpp/sugar/functions/ifelse.h>
#include <Rcpp/sugar/functions/pmin.h>
#include <Rcpp/sugar/functions/pmax.h>
#include <Rcpp/sugar/functions/clamp.h>
#include <Rcpp/sugar/functions/min.h>
#include <Rcpp/sugar/functions/max.h>
#include <Rcpp/sugar/functions/range.h>
#include <Rcpp/sugar/functions/sign.h>
#include <Rcpp/sugar/functions/diff.h>
#include <Rcpp/sugar/functions/pow.h>
#include <Rcpp/sugar/functions/rep.h>
```

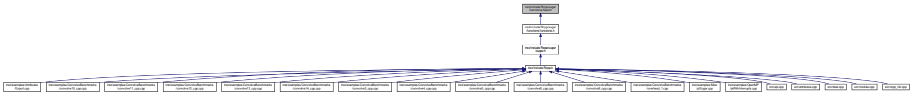
```
#include <Rcpp/sugar/functions/rep_len.h>
#include <Rcpp/sugar/functions/rep_each.h>
#include <Rcpp/sugar/functions/rev.h>
#include <Rcpp/sugar/functions/head.h>
#include <Rcpp/sugar/functions/tail.h>
#include <Rcpp/sugar/functions/sum.h>
#include <Rcpp/sugar/functions/mean.h>
#include <Rcpp/sugar/functions/var.h>
#include <Rcpp/sugar/functions/sd.h>
#include <Rcpp/sugar/functions/cumsum.h>
#include <Rcpp/sugar/functions/which_min.h>
#include <Rcpp/sugar/functions/which_max.h>
#include <Rcpp/sugar/functions/unique.h>
#include <Rcpp/sugar/functions/match.h>
#include <Rcpp/sugar/functions/table.h>
#include <Rcpp/sugar/functions/duplicated.h>
#include <Rcpp/sugar/functions/self_match.h>
#include <Rcpp/sugar/functions/setdiff.h>
#include <Rcpp/sugar/functions/strings/strings.h>
#include <Rcpp/sugar/functions/cumprod.h>
#include <Rcpp/sugar/functions/cummin.h>
#include <Rcpp/sugar/functions/cummax.h>
#include <Rcpp/sugar/functions/median.h>
#include <Rcpp/sugar/functions/cbind.h>
#include <Rcpp/sugar/functions/rowSums.h>
#include <Rcpp/sugar/functions/sample.h>
```

This graph shows which files directly or indirectly include this file:



7.247 inst/include/Rcpp/sugar/functions/head.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Head< RTYPE, NA, T >

Namespaces

- Rcpp

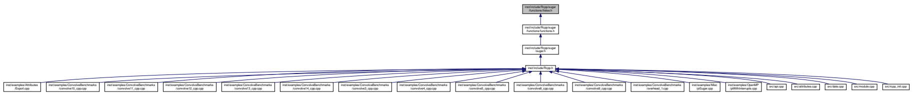
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Head< RTYPE, NA, T > Rcpp::head (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)

7.248 inst/include/Rcpp/sugar/functions/ifelse.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/ifelse.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >
- class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >
- class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >
- class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >
- class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >
- class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

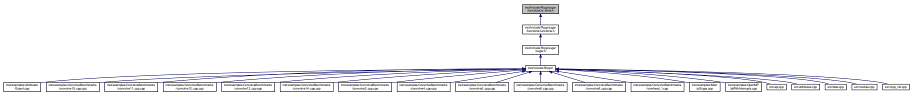
Functions

- template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
  sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, typename traits::storage_type< RTYPE >↩
  ::type lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
  sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &lhs, typename traits::storage_type< RTYPE >::type rhs)

- template<bool COND_NA, typename COND_T >
  sugar::IfElse_Primitive_Primitive< REALSXP, COND_NA, COND_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, double lhs, double rhs)
- template<bool COND_NA, typename COND_T >
  sugar::IfElse_Primitive_Primitive< INTSXP, COND_NA, COND_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, int lhs, int rhs)
- template<bool COND_NA, typename COND_T >
  sugar::IfElse_Primitive_Primitive< CPLXSXP, COND_NA, COND_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, Rcomplex lhs, Rcomplex rhs)
- template<bool COND_NA, typename COND_T >
  sugar::IfElse_Primitive_Primitive< LGLSXP, COND_NA, COND_T > Rcpp::ifelse (const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > &cond, bool lhs, bool rhs)

7.249 inst/include/Rcpp/sugar/functions/is_finite.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/is_finite.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >

Namespaces

- Rcpp

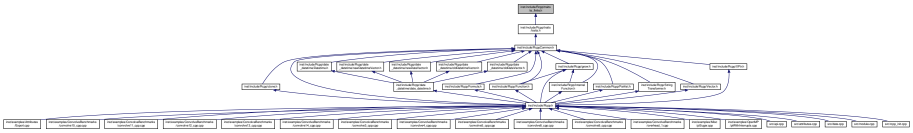
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::IsFinite< RTYPE, NA, T > Rcpp::is_finite (const Rcpp::VectorBase< RTYPE, NA, T > &t)

7.250 inst/include/Rcpp/traits/is_finite.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/is_finite.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

    Rcpp API.
- Rcpp::traits

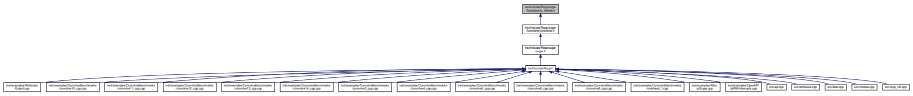
    traits used to dispatch wrap

Functions

- template<int RTYPE>
  bool Rcpp::traits::is_finite (typename storage_type< RTYPE >::type)
- template<> bool Rcpp::traits::is_finite< INTSXP > (int x)
- template<> bool Rcpp::traits::is_finite< REALSXP > (double x)
- template<> bool Rcpp::traits::is_finite< CPLXSXP > (Rcomplex x)
- template<> bool Rcpp::traits::is_finite< STRSXP > (SEXP)
- template<> bool Rcpp::traits::is_finite< LGLSXP > (int x)

7.251 inst/include/Rcpp/sugar/functions/is_infinite.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/is_infinite.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >

Namespaces

- Rcpp

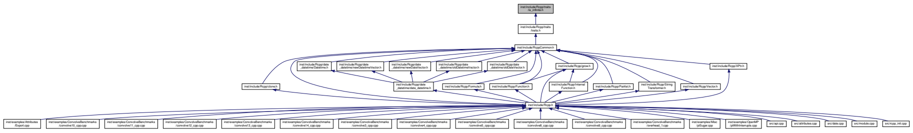
    Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::IsInfinite< RTYPE, NA, T > Rcpp::is_infinite (const Rcpp::VectorBase< RTYPE, NA, T > &t)

7.252 inst/include/Rcpp/traits/is_infinite.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

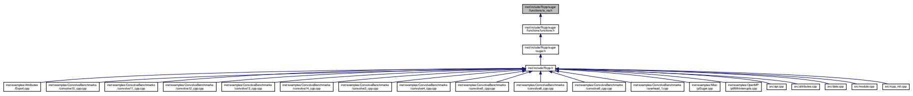
  traits used to dispatch wrap

Functions

- template< int RTYPE >
  bool Rcpp::traits::is_infinite (typename storage_type< RTYPE >::type)
- template<> bool Rcpp::traits::is_infinite< REALSXP > (double x)
- template<> bool Rcpp::traits::is_infinite< CPLXSXP > (Rcomplex x)

7.253 inst/include/Rcpp/sugar/functions/is_na.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >
- class Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >
- class Rcpp::sugar::IsNa_Vector_is_na< T >

Namespaces

- Rcpp

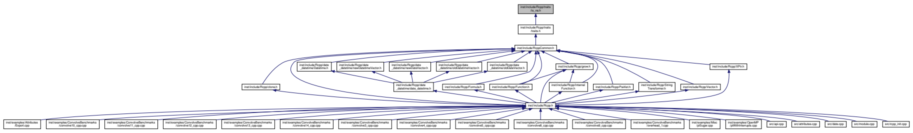
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::IsNa< RTYPE, NA, T > Rcpp::is_na (const Rcpp::VectorBase< RTYPE, NA, T > &t)
- sugar::IsNa_Vector_is_na< oldDatetimeVector > Rcpp::is_na (const oldDatetimeVector &x)
- sugar::IsNa_Vector_is_na< oldDateVector > Rcpp::is_na (const oldDateVector &x)
- sugar::IsNa_Vector_is_na< NumericVector > Rcpp::is_na (newDatetimeVector &x)
- sugar::IsNa_Vector_is_na< NumericVector > Rcpp::is_na (newDateVector &x)

7.254 inst/include/Rcpp/traits/is_na.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

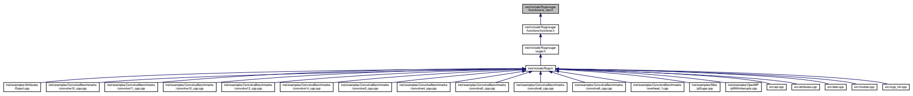
  traits used to dispatch wrap

Functions

- template<int RTYPE>
  bool Rcpp::traits::is_na (typename storage_type< RTYPE >::type)
- template<> bool Rcpp::traits::is_na< INTSXP > (int x)
- template<> bool Rcpp::traits::is_na< REALSXP > (double x)
- template<> bool Rcpp::traits::is_na< CPLXSXP > (Rcomplex x)
- template<> bool Rcpp::traits::is_na< STRSXP > (SEXP x)
- template<> bool Rcpp::traits::is_na< LGLSXP > (int x)

7.255 inst/include/Rcpp/sugar/functions/is_nan.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >

Namespaces

- Rcpp

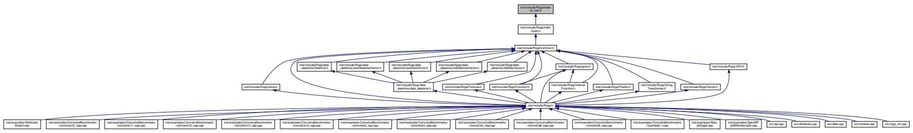
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::IsNaN< RTYPE, NA, T > Rcpp::is_nan (const Rcpp::VectorBase< RTYPE, NA, T > &t)

7.256 inst/include/Rcpp/traits/is_nan.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

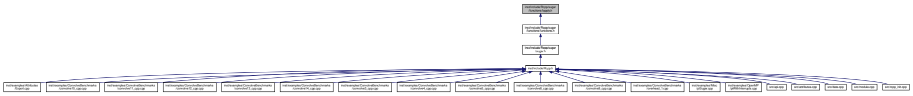
  traits used to dispatch wrap

Functions

- template<int RTYPE>
  bool Rcpp::traits::is_nan (typename storage_type< RTYPE >::type)
- template<> bool Rcpp::traits::is_nan< REALSXP > (double x)
- template<> bool Rcpp::traits::is_nan< CPLXSXP > (Rcomplex x)

7.257 inst/include/Rcpp/sugar/functions/lapply.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/lapply.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Lapply< RTYPE, NA, T, Function >

Namespaces

- Rcpp

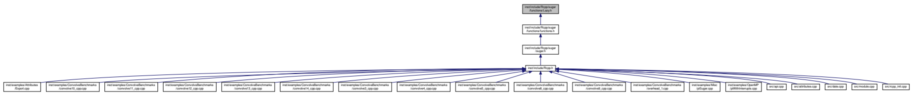
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T , typename Function >
  sugar::Lapply< RTYPE, NA, T, Function > Rcpp::lapply (const Rcpp::VectorBase< RTYPE, NA, T > &t, Function fun)

7.258 inst/include/Rcpp/sugar/functions/Lazy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/Lazy.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Lazy< T, EXPR >

Namespaces

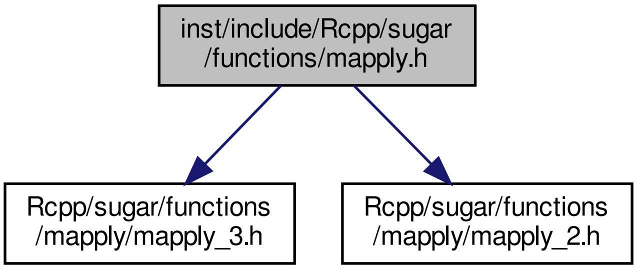
- Rcpp

  Rcpp API.
- Rcpp::sugar

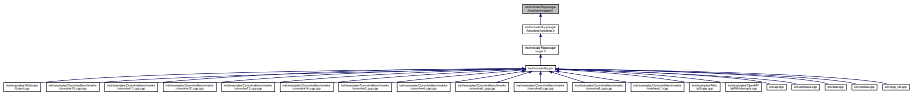
7.259 inst/include/Rcpp/sugar/functions/mapply.h File Reference

```
#include <Rcpp/sugar/functions/mapply/mapply_3.h>
#include <Rcpp/sugar/functions/mapply/mapply_2.h>
```

Include dependency graph for mapply.h:

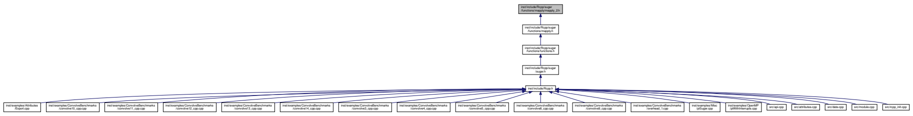


This graph shows which files directly or indirectly include this file:



7.260 inst/include/Rcpp/sugar/functions/mapply/mapply_2.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >
- class Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >
- class Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >

Namespaces

- Rcpp

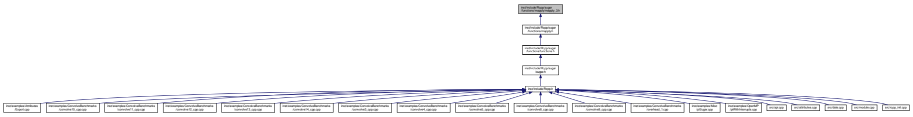
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
  sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > Rcpp::mapply (const Rcpp::VectorBase< RTYPE, NA_1, T_1 > &t1, const Rcpp::VectorBase< RTYPE, NA_2, T_2 > &t2, Function fun)
- template< int RTYPE, bool NA_1, typename T_1 , typename Function >
  sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, double, Function > Rcpp::mapply (const Rcpp::VectorBase< RTYPE, NA_1, T_1 > &t1, double t2, Function fun)
- template< int RTYPE, bool NA_2, typename T_2 , typename Function >
  sugar::Mapply_2_Primitive_Vector< RTYPE, double, NA_2, T_2, Function > Rcpp::mapply (double t1, const Rcpp::VectorBase< RTYPE, NA_2, T_2 > &t2, Function fun)

7.261 inst/include/Rcpp/sugar/functions/mapply/mapply_3.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >

Namespaces

- Rcpp

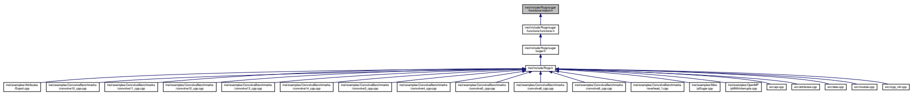
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
  sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > Rcpp::mapply (const Rcpp::VectorBase< RTYPE_1, NA_1, T_1 > &t1, const Rcpp::VectorBase< RTYPE↩_2, NA_2, T_2 > &t2, const Rcpp::VectorBase< RTYPE_3, NA_3, T_3 > &t3, Function fun)

7.262 inst/include/Rcpp/sugar/functions/match.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

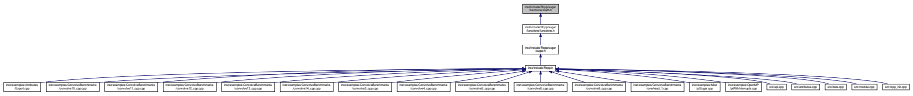
  Rcpp API.

Functions

- template< int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
  IntegerVector Rcpp::match (const VectorBase< RTYPE, NA, T > &x, const VectorBase< RTYPE, RHS_NA, RHS_T > &table_)

7.263 inst/include/Rcpp/sugar/functions/math.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Functions

- double Rcpp::internal::factorial (double x)
- double Rcpp::internal::lfactorial (double x)
- VECTORIZED_MATH_1 (trunc, ::Rf_ftrunc) SUGAR_BLOCK_2(round

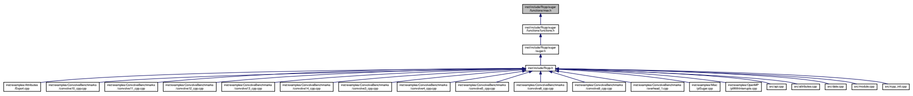
7.263.1 Function Documentation

7.263.1.1 VECTORIZED_MATH_1()

```
VECTORIZED_MATH_1 (
             trunc ,
             ::Rf_ftrunc  )
```

7.264 inst/include/Rcpp/sugar/functions/max.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/max.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Max< RTYPE, NA, T >
- class Rcpp::sugar::Max< RTYPE, false, T >

Namespaces

- Rcpp

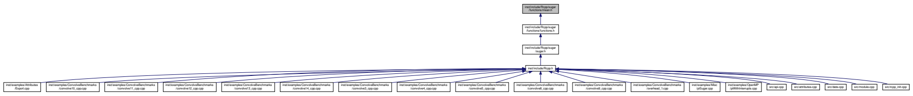
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Max< RTYPE, NA, T > Rcpp::max (const VectorBase< RTYPE, NA, T > &x)

7.265 inst/include/Rcpp/sugar/functions/mean.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Mean< RTYPE, NA, T >
- class Rcpp::sugar::Mean< CPLXSXP, NA, T >
- class Rcpp::sugar::Mean< LGLSXP, NA, T >
- class Rcpp::sugar::Mean< INTSXP, NA, T >

Namespaces

- Rcpp

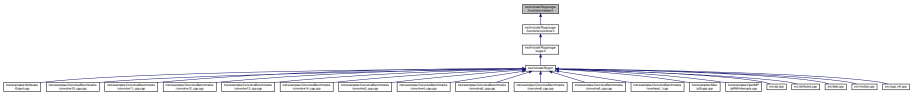
  Rcpp API.
- Rcpp::sugar

Functions

- template< bool NA, typename T >
  sugar::Mean< REALSXP, NA, T > Rcpp::mean (const VectorBase< REALSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Mean< INTSXP, NA, T > Rcpp::mean (const VectorBase< INTSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Mean< CPLXSXP, NA, T > Rcpp::mean (const VectorBase< CPLXSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Mean< LGLSXP, NA, T > Rcpp::mean (const VectorBase< LGLSXP, NA, T > &t)

7.266 inst/include/Rcpp/sugar/functions/median.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::sugar::median_detail::result< RTYPE >
- struct Rcpp::sugar::median_detail::result< INTSXP >
- struct Rcpp::sugar::median_detail::result< STRSXP >
- class Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >
- class Rcpp::sugar::Median< RTYPE, NA, T, true >
- class Rcpp::sugar::Median< RTYPE, false, T, NA_RM >
- class Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >
- class Rcpp::sugar::Median< STRSXP, NA, T, true >
- class Rcpp::sugar::Median< STRSXP, false, T, true >

Namespaces

- Rcpp

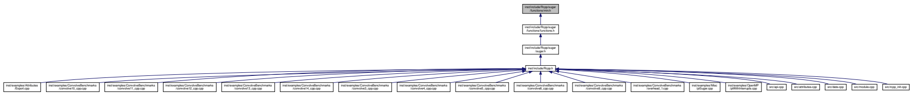
  Rcpp API.
- Rcpp::sugar
- Rcpp::sugar::median_detail

Functions

- template<typename T >
  bool Rcpp::sugar::median_detail::less (T lhs, T rhs)
- template<> bool Rcpp::sugar::median_detail::less< Rcomplex > (Rcomplex lhs, Rcomplex rhs)
- double Rcpp::sugar::median_detail::half (double lhs)
- double Rcpp::sugar::median_detail::half (int lhs)
- Rcomplex Rcpp::sugar::median_detail::half (Rcomplex lhs)
- template<int RTYPE, bool NA, typename T >
  sugar::median_detail::result< RTYPE >::type Rcpp::median (const Rcpp::VectorBase< RTYPE, NA, T > &x, bool na_rm=false)

7.267 inst/include/Rcpp/sugar/functions/min.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Min< RTYPE, NA, T >
- class Rcpp::sugar::Min< RTYPE, false, T >

Namespaces

- Rcpp

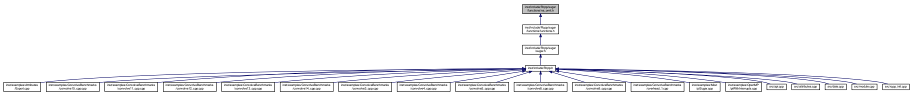
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Min< RTYPE, NA, T > Rcpp::min (const VectorBase< RTYPE, NA, T > &x)

7.268 inst/include/Rcpp/sugar/functions/na_omit.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

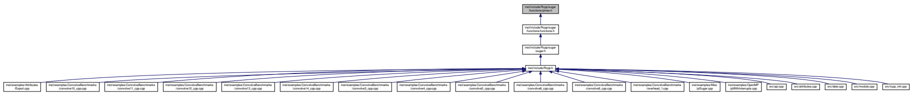
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE, bool NA, typename T >
  Vector< RTYPE > Rcpp::sugar::na_omit_impl (const T &x, Rcpp::traits::false_type)
- template< int RTYPE, bool NA, typename T >
  Vector< RTYPE > Rcpp::sugar::na_omit_impl (const T &x, Rcpp::traits::true_type)
- template< int RTYPE, bool NA, typename T >
  Vector< RTYPE > Rcpp::na_omit (const VectorBase< RTYPE, NA, T > &t)

7.269 inst/include/Rcpp/sugar/functions/pmax.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::sugar::pmax_op< REALSXP, true, true >
- struct Rcpp::sugar::pmax_op< REALSXP, true, false >
- struct Rcpp::sugar::pmax_op< REALSXP, false, true >
- struct Rcpp::sugar::pmax_op< REALSXP, false, false >
- struct Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >
- class Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >
- class Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >
- class Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >

Namespaces

- Rcpp

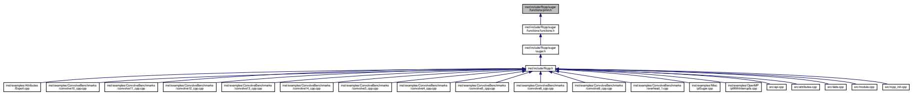
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::pmax (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Rcpp::pmax (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool RHS_NA, typename RHS_T >
  sugar::Pmax_Vector_Primitive< RTYPE, RHS_NA, RHS_T > Rcpp::pmax (typename Rcpp::traits::storage_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.270 inst/include/Rcpp/sugar/functions/pmin.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/pmin.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::sugar::pmin_op< REALSXP, true, true >
- struct Rcpp::sugar::pmin_op< REALSXP, true, false >
- struct Rcpp::sugar::pmin_op< REALSXP, false, true >
- struct Rcpp::sugar::pmin_op< REALSXP, false, false >
- struct Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >
- class Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >
- class Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >
- class Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >

Namespaces

- Rcpp

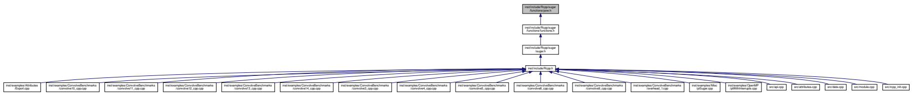
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::pmin (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Rcpp::pmin (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool RHS_NA, typename RHS_T >
  sugar::Pmin_Vector_Primitive< RTYPE, RHS_NA, RHS_T > Rcpp::pmin (typename Rcpp::traits::storage_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.271 inst/include/Rcpp/sugar/functions/pow.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/pow.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >
- class Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >
- class Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >

Namespaces

- Rcpp

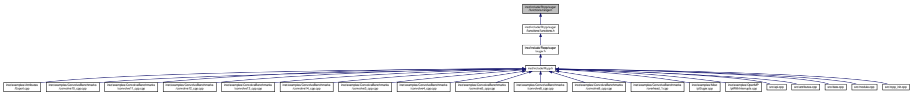
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
  sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > Rcpp::pow (const VectorBase< RTYPE, NA, T > &t, EXPONENT_TYPE exponent)

7.272 inst/include/Rcpp/sugar/functions/range.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/range.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Range< RTYPE, NA, T >
- class Rcpp::sugar::Range< RTYPE, false, T >

Namespaces

- Rcpp

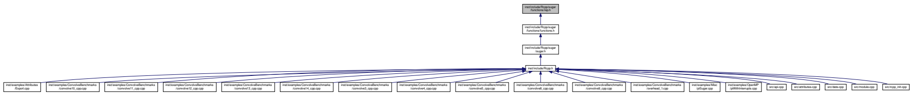
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Range< RTYPE, NA, T > Rcpp::range (const VectorBase< RTYPE, NA, T > &x)

7.273 inst/include/Rcpp/sugar/functions/rep.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Rep< RTYPE, NA, T >
- class Rcpp::sugar::Rep_Single< T >

Namespaces

- Rcpp

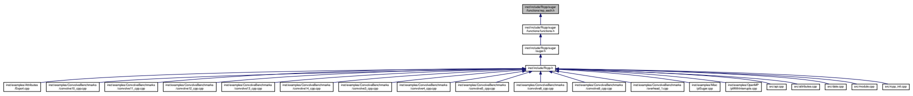
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Rep< RTYPE, NA, T > Rcpp::rep (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)
- sugar::Rep_Single< double > Rcpp::rep (const double &x, R_xlen_t n)
- sugar::Rep_Single< int > Rcpp::rep (const int &x, R_xlen_t n)
- sugar::Rep_Single< Rbyte > Rcpp::rep (const Rbyte &x, R_xlen_t n)
- sugar::Rep_Single< Rcomplex > Rcpp::rep (const Rcomplex &x, R_xlen_t n)
- sugar::Rep_Single< bool > Rcpp::rep (const bool &x, R_xlen_t n)

7.274 inst/include/Rcpp/sugar/functions/rep_each.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Rep_each< RTYPE, NA, T >

Namespaces

- Rcpp

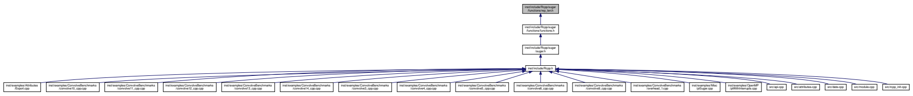
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Rep_each< RTYPE, NA, T > Rcpp::rep_each (const VectorBase< RTYPE, NA, T > &t, R_xlen_t times)

7.275 inst/include/Rcpp/sugar/functions/rep_len.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Rep_len< RTYPE, NA, T >

Namespaces

- Rcpp

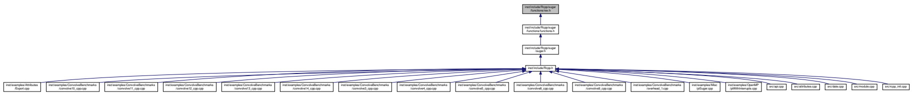
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Rep_len< RTYPE, NA, T > Rcpp::rep_len (const VectorBase< RTYPE, NA, T > &t, R_xlen_t len)

7.276 inst/include/Rcpp/sugar/functions/rev.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/rev.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Rev< RTYPE, NA, T >

Namespaces

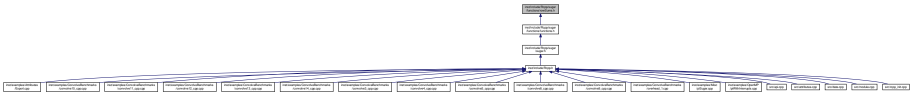
- Rcpp
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Rev< RTYPE, NA, T > Rcpp::rev (const VectorBase< RTYPE, NA, T > &t)

7.277 inst/include/Rcpp/sugar/functions/rowSums.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/rowSums.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::sugar::detail::RowSumsReturn< RTYPE >
- struct Rcpp::sugar::detail::RowSumsReturn< LGLSXP >
- struct Rcpp::sugar::detail::ColSumsReturn< RTYPE >
- struct Rcpp::sugar::detail::RowMeansReturn< RTYPE >
- struct Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >
- struct Rcpp::sugar::detail::ColMeansReturn< RTYPE >
- class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >
- class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >
- class Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >
- class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >
- class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >
- class Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >
- class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >
- class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >
- class Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >
- class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >
- class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >
- class Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar
- Rcpp::sugar::detail

Macros

- #define ROW_SUMS_IMPL_KEEPNA(__RTYPE__)
- #define ROW_SUMS_IMPL_RMNA(__RTYPE__)
- #define COL_SUMS_IMPL_KEEPNA(__RTYPE__)
- #define COL_SUMS_IMPL_RMNA(__RTYPE__)
- #define ROW_MEANS_IMPL_KEEPNA(__RTYPE__)
- #define ROW_MEANS_IMPL_RMNA(__RTYPE__)
- #define COL_MEANS_IMPL_KEEPNA(__RTYPE__)
- #define COL_MEANS_IMPL_RMNA(__RTYPE__)

Functions

- bool Rcpp::sugar::detail::check_na (double x)
- bool Rcpp::sugar::detail::check_na (int x)
- bool Rcpp::sugar::detail::check_na (Rboolean x)
- bool Rcpp::sugar::detail::check_na (SEXP x)
- bool Rcpp::sugar::detail::check_na (Rcomplex x)
- void Rcpp::sugar::detail::incr (double ∗lhs, double rhs)
- void Rcpp::sugar::detail::incr (int ∗lhs, int rhs)
- void Rcpp::sugar::detail::incr (Rcomplex ∗lhs, const Rcomplex &rhs)
- void Rcpp::sugar::detail::div (double ∗lhs, R_xlen_t rhs)
- void Rcpp::sugar::detail::div (Rcomplex ∗lhs, R_xlen_t rhs)
- void Rcpp::sugar::detail::set_nan (double ∗x)
- void Rcpp::sugar::detail::set_nan (Rcomplex ∗x)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::RowSumsReturn< RTYPE >::type Rcpp::rowSums (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::ColSumsReturn< RTYPE >::type Rcpp::colSums (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::RowMeansReturn< RTYPE >::type Rcpp::rowMeans (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)
- template<int RTYPE, bool NA, typename T >
  sugar::detail::ColMeansReturn< RTYPE >::type Rcpp::colMeans (const MatrixBase< RTYPE, NA, T > &x, bool na_rm=false)

7.277.1 Macro Definition Documentation

7.277.1.1 COL_MEANS_IMPL_KEEPNA

```
#define COL_MEANS_IMPL_KEEPNA(
              __RTYPE__ )
```

Definition at line 753 of file rowSums.h.

7.277.1.2 COL_MEANS_IMPL_RMNA

```
#define COL_MEANS_IMPL_RMNA(
              __RTYPE__ )
```

Definition at line 861 of file rowSums.h.

7.277.1.3 COL_SUMS_IMPL_KEEPNA

```
#define COL_SUMS_IMPL_KEEPNA(
             __RTYPE__ )
```

Definition at line 353 of file rowSums.h.

7.277.1.4 COL_SUMS_IMPL_RMNA

```
#define COL_SUMS_IMPL_RMNA(
             __RTYPE__ )
```

Definition at line 448 of file rowSums.h.

7.277.1.5 ROW_MEANS_IMPL_KEEPNA

```
#define ROW_MEANS_IMPL_KEEPNA(
             __RTYPE__ )
```

Definition at line 544 of file rowSums.h.

7.277.1.6 ROW_MEANS_IMPL_RMNA

```
#define ROW_MEANS_IMPL_RMNA(
             __RTYPE__ )
```

Definition at line 652 of file rowSums.h.

7.277.1.7 ROW_SUMS_IMPL_KEEPNA

```
#define ROW_SUMS_IMPL_KEEPNA(
             __RTYPE__ )
```

Definition at line 169 of file rowSums.h.

7.277.1.8 ROW_SUMS_IMPL_RMNA

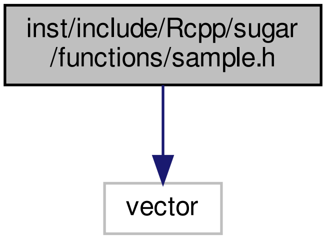
```
#define ROW_SUMS_IMPL_RMNA(
              __RTYPE__ )
```

Definition at line 264 of file rowSums.h.

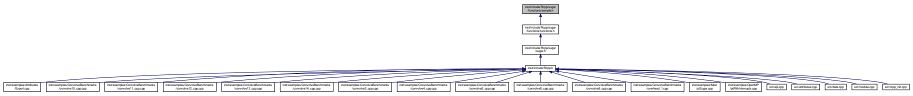
7.278 inst/include/Rcpp/sugar/functions/sample.h File Reference

```
#include <vector>
```

Include dependency graph for sample.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.
- Rcpp::sugar

Typedefs

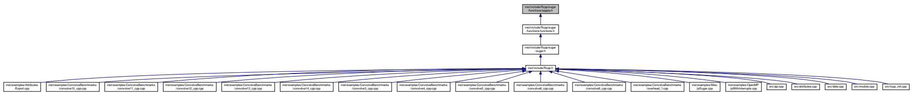
- typedef Nullable< Vector< REALSXP > > Rcpp::sugar::probs_t

Functions

- void Rcpp::sugar::Normalize (Vector< REALSXP > &p, int require_k, bool replace)
- Vector< INTSXP > Rcpp::sugar::SampleReplace (Vector< REALSXP > &p, int n, int k, bool one_based)
- template<int RTYPE>
  Vector< RTYPE > Rcpp::sugar::SampleReplace (Vector< REALSXP > &p, int k, const Vector< RTYPE > &ref)
- Vector< INTSXP > Rcpp::sugar::WalkerSample (const Vector< REALSXP > &p, int n, int nans, bool one_based)
- template<int RTYPE>
  Vector< RTYPE > Rcpp::sugar::WalkerSample (const Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > Rcpp::sugar::SampleNoReplace (Vector< REALSXP > &p, int n, int nans, bool one_based)
- template<int RTYPE>
  Vector< RTYPE > Rcpp::sugar::SampleNoReplace (Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > Rcpp::sugar::EmpiricalSample (int n, int size, bool replace, bool one_based)
- template<int RTYPE>
  Vector< RTYPE > Rcpp::sugar::EmpiricalSample (int size, bool replace, const Vector< RTYPE > &ref)
- Vector< INTSXP > Rcpp::sample (int n, int size, bool replace=false, sugar::probs_t probs=R_NilValue, bool one_based=true)
- template<int RTYPE>
  Vector< RTYPE > Rcpp::sample (const Vector< RTYPE > &x, int size, bool replace=false, sugar::probs_↩t probs=R_NilValue)

7.279 inst/include/Rcpp/sugar/functions/sapply.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/sapply.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::sugar::sapply_application_result_of< Function, SugarExpression >
- class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >
- class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >

Namespaces

- Rcpp

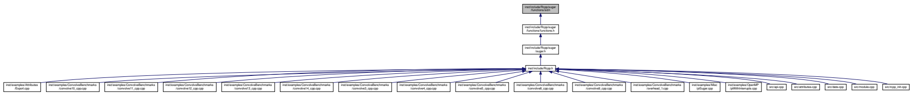
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE, bool NA, typename T , typename Function >
  sugar::Sapply< RTYPE, NA, T, Function, traits::same_type< typename ::Rcpp::sugar::sapply_application_result_of< Function, T >::type, typename Rcpp::traits::storage_type< traits::r_sexptype_traits< typename ::Rcpp::sugar::sapply_application_r Function, T >::type >::rtype >::type >::value > Rcpp::sapply (const Rcpp::VectorBase< RTYPE, NA, T > &t, Function fun)

7.280 inst/include/Rcpp/sugar/functions/sd.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Sd< RTYPE, NA, T >

Namespaces

- Rcpp

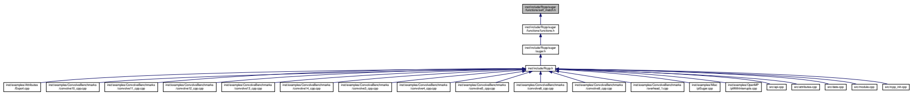
  Rcpp API.
- Rcpp::sugar

Functions

- template< bool NA, typename T >
  sugar::Sd< REALSXP, NA, T > Rcpp::sd (const VectorBase< REALSXP, NA, T > &t)

7.281 inst/include/Rcpp/sugar/functions/self_match.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SelfInserter< HASH, STORAGE >
- class Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >

Namespaces

- Rcpp

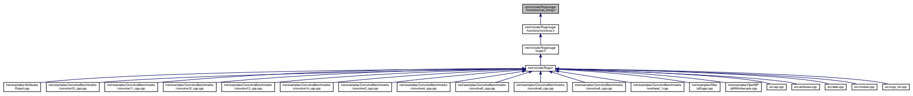
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  IntegerVector Rcpp::self_match (const VectorBase< RTYPE, NA, T > &x)

7.282 inst/include/Rcpp/sugar/functions/seq_along.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/seq_along.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::SeqLen

Namespaces

- Rcpp

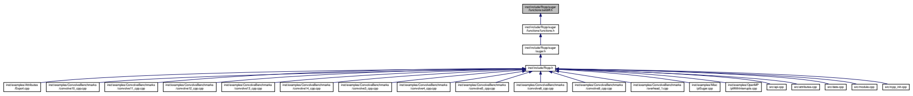
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::SeqLen Rcpp::seq_along (const Rcpp::VectorBase< RTYPE, NA, T > &t)
- sugar::SeqLen Rcpp::seq_len (const size_t &n)
- Range Rcpp::seq (R_xlen_t start, R_xlen_t end)

7.283 inst/include/Rcpp/sugar/functions/setdiff.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/setdiff.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::RemoveFromSet< SET >
- class Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >

Namespaces

- Rcpp

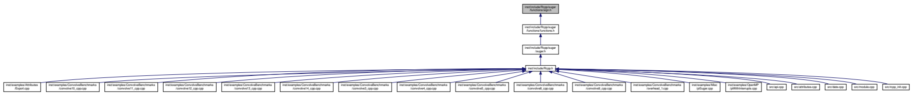
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Vector< RTYPE > Rcpp::setdiff (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  bool Rcpp::setequal (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Vector< RTYPE > Rcpp::intersect (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Vector< RTYPE > Rcpp::union_ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.284 inst/include/Rcpp/sugar/functions/sign.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/sign.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::sign__impl< NA, RTYPE >
- class Rcpp::sugar::sign__impl< false, RTYPE >
- class Rcpp::sugar::Sign< RTYPE, NA, T >

Namespaces

- Rcpp

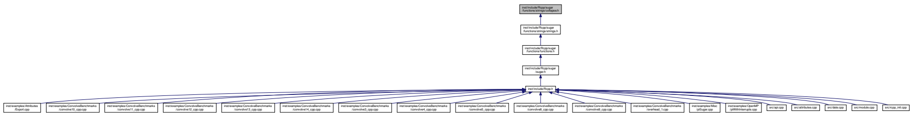
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Sign< INTSXP, NA, T > Rcpp::sign (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Sign< REALSXP, NA, T > Rcpp::sign (const VectorBase< REALSXP, NA, T > &t)

7.285 inst/include/Rcpp/sugar/functions/strings/collapse.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/strings/collapse.h
inst/include/Rcpp/sugar/functions/strings/strings.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

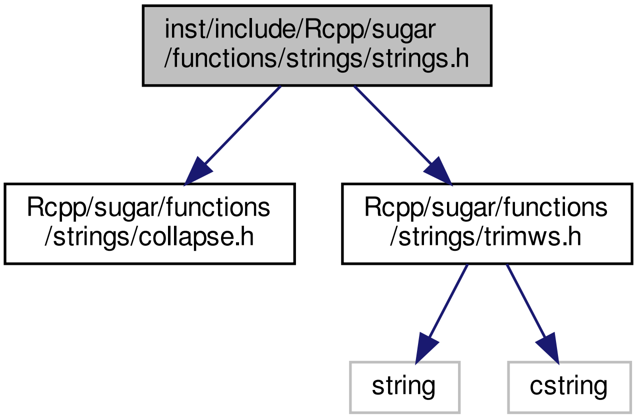
Functions

- template<typename Iterator >
  String Rcpp::sugar::collapse__impl (Iterator it, R_xlen_t n)
- template<bool NA, typename T >
  String Rcpp::collapse (const VectorBase< STRSXP, NA, T > &vec)

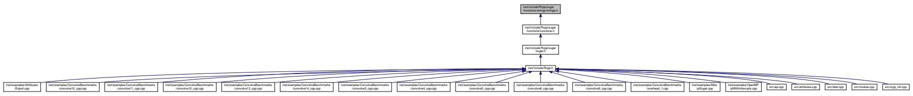
7.286 inst/include/Rcpp/sugar/functions/strings/strings.h File Reference

```
#include <Rcpp/sugar/functions/strings/collapse.h>
#include <Rcpp/sugar/functions/strings/trimws.h>
```

Include dependency graph for strings.h:



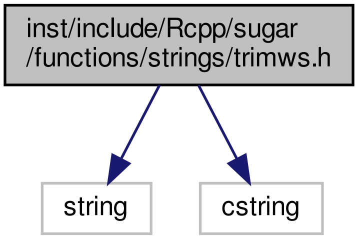
This graph shows which files directly or indirectly include this file:



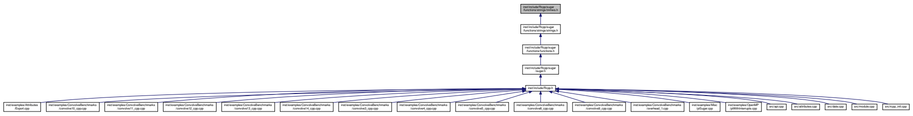
7.287 inst/include/Rcpp/sugar/functions/strings/trimws.h File Reference

```
#include <string>
#include <cstring>
```

Include dependency graph for trimws.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

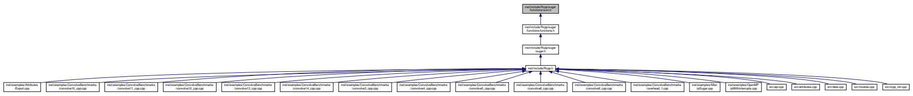
  Rcpp API.
- Rcpp::sugar
- Rcpp::sugar::detail

Functions

- bool Rcpp::sugar::detail::isws (const char c)
- const char ∗ Rcpp::sugar::detail::trim_left (const char ∗str)
- const char ∗ Rcpp::sugar::detail::trim_right (const char ∗str, R_len_t sz, std::string ∗buff)
- const char ∗ Rcpp::sugar::detail::trim_both (const char ∗str, R_len_t sz, std::string ∗buff)
- Vector< STRSXP > Rcpp::trimws (const Vector< STRSXP > &x, const char ∗which="both")
- Matrix< STRSXP > Rcpp::trimws (const Matrix< STRSXP > &x, const char ∗which="both")
- String Rcpp::trimws (const String &str, const char ∗which="both")

7.288 inst/include/Rcpp/sugar/functions/sum.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Sum< RTYPE, NA, T >
- class Rcpp::sugar::Sum< REALSXP, NA, T >
- class Rcpp::sugar::Sum< RTYPE, false, T >

Namespaces

- Rcpp

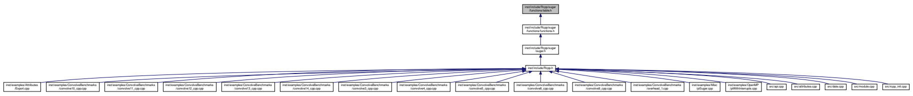
  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  sugar::Sum< INTSXP, NA, T > Rcpp::sum (const VectorBase< INTSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Sum< REALSXP, NA, T > Rcpp::sum (const VectorBase< REALSXP, NA, T > &t)
- template<bool NA, typename T >
  sugar::Sum< LGLSXP, NA, T > Rcpp::sum (const VectorBase< LGLSXP, NA, T > &t)

7.289 inst/include/Rcpp/sugar/functions/table.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/table.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::CountInserter< HASH, STORAGE >
- class Rcpp::sugar::Grabber< HASH, RTYPE >
- class Rcpp::sugar::Table< RTYPE, TABLE_T >

Namespaces

- Rcpp

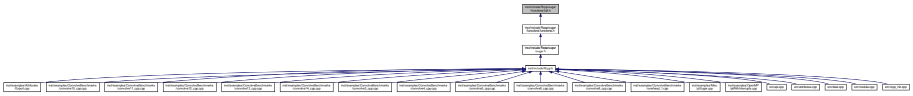
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  IntegerVector Rcpp::table (const VectorBase< RTYPE, NA, T > &x)

7.290 inst/include/Rcpp/sugar/functions/tail.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Tail< RTYPE, NA, T >

Namespaces

- Rcpp

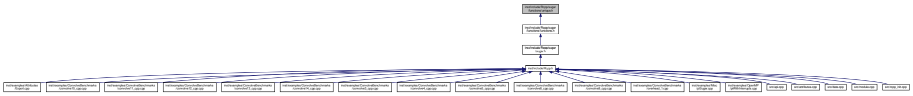
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::Tail< RTYPE, NA, T > Rcpp::tail (const VectorBase< RTYPE, NA, T > &t, R_xlen_t n)

7.291 inst/include/Rcpp/sugar/functions/unique.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::InSet< HASH >
- class Rcpp::sugar::In< RTYPE, TABLE_T >

Namespaces

- Rcpp

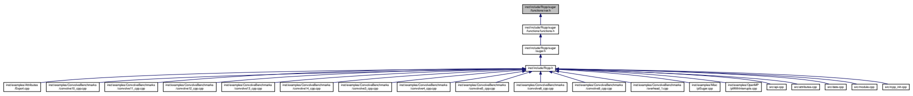
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE, bool NA, typename T >
  Vector< RTYPE > Rcpp::unique (const VectorBase< RTYPE, NA, T > &t)
- template< int RTYPE, bool NA, typename T >
  Vector< RTYPE > Rcpp::sort_unique (const VectorBase< RTYPE, NA, T > &t, bool decreasing=false)
- template< int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
  LogicalVector Rcpp::in (const VectorBase< RTYPE, NA, T > &x, const VectorBase< RTYPE, RHS_NA, RHS_T > &table)

7.292 inst/include/Rcpp/sugar/functions/var.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/functions/var.h
inst/include/Rcpp/sugar/functions/functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Var< RTYPE, NA, T >
- class Rcpp::sugar::Var< CPLXSXP, NA, T >

Namespaces

- Rcpp

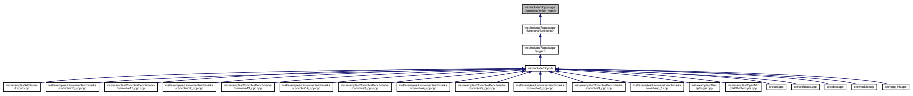
  Rcpp API.
- Rcpp::sugar

Functions

- template< bool NA, typename T >
  sugar::Var< REALSXP, NA, T > Rcpp::var (const VectorBase< REALSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Var< INTSXP, NA, T > Rcpp::var (const VectorBase< INTSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Var< LGLSXP, NA, T > Rcpp::var (const VectorBase< LGLSXP, NA, T > &t)
- template< bool NA, typename T >
  sugar::Var< CPLXSXP, NA, T > Rcpp::var (const VectorBase< CPLXSXP, NA, T > &t)

7.293 inst/include/Rcpp/sugar/functions/which_max.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::WhichMax< RTYPE, NA, T >
- class Rcpp::sugar::WhichMax< RTYPE, false, T >

Namespaces

- Rcpp

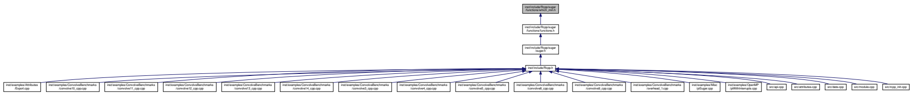
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  R_xlen_t Rcpp::which_max (const VectorBase< RTYPE, NA, T > &t)

7.294 inst/include/Rcpp/sugar/functions/which_min.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::WhichMin< RTYPE, NA, T >
- class Rcpp::sugar::WhichMin< RTYPE, false, T >

Namespaces

- Rcpp

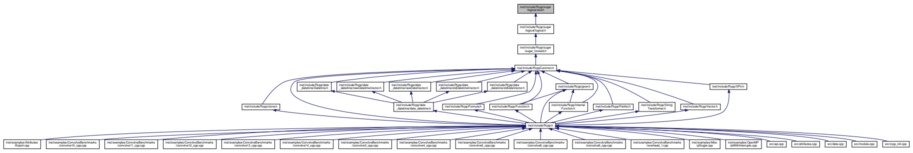
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  R_xlen_t Rcpp::which_min (const VectorBase< RTYPE, NA, T > &t)

7.295 inst/include/Rcpp/sugar/logical/and.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >
- class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Functions

- template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > operator&& (const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs, const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &rhs)
- template<bool LHS_NA, typename LHS_T >
  Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > operator&& (const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs, bool rhs)
- template<bool LHS_NA, typename LHS_T >
  Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > operator&& (bool rhs, const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs)
- template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > operator& (const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > &rhs)

7.295.1 Function Documentation

7.295.1.1 operator&()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression<LHS_NA,LHS_T,RHS_NA,RHS_T> operator& (
            const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > & lhs,
            const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 316 of file and.h.

7.295.1.2 operator&&() [1/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool<LHS_NA,LHS_T> operator&& (
            bool rhs,
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs )  [inline]
```

Definition at line 306 of file and.h.

7.295.1.3 operator&&() [2/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool<LHS_NA,LHS_T> operator&& (
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
            bool rhs )  [inline]
```

Definition at line 297 of file and.h.

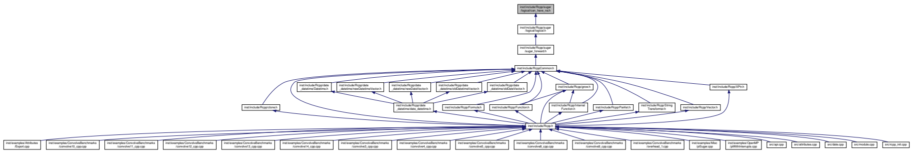
7.295.1.4 operator&&() [3/3]

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_T,RHS_NA,RHS_T> operator&& (
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & rhs )  [inline]
```

Definition at line 288 of file and.h.

7.296 inst/include/Rcpp/sugar/logical/can_have_na.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::can_have_na< T >

Namespaces

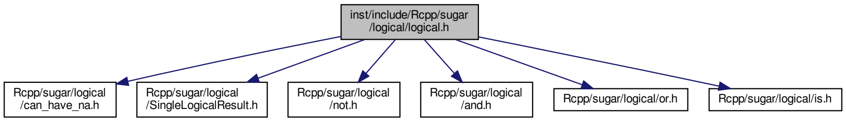
- Rcpp

  Rcpp API.

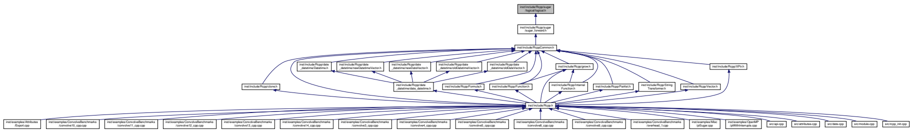
7.297 inst/include/Rcpp/sugar/logical/logical.h File Reference

```
#include <Rcpp/sugar/logical/can_have_na.h>
#include <Rcpp/sugar/logical/SingleLogicalResult.h>
#include <Rcpp/sugar/logical/not.h>
#include <Rcpp/sugar/logical/and.h>
#include <Rcpp/sugar/logical/or.h>
#include <Rcpp/sugar/logical/is.h>
```

Include dependency graph for logical.h:

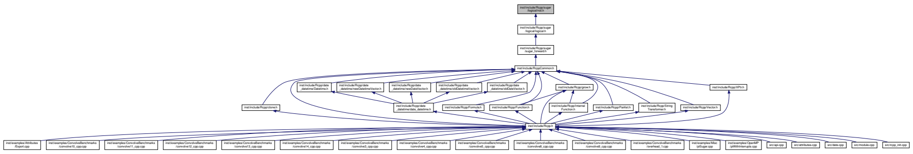


This graph shows which files directly or indirectly include this file:



7.298 inst/include/Rcpp/sugar/logical/not.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::sugar::negate< NA >
- struct Rcpp::sugar::negate< false >
- class Rcpp::sugar::Negate_SingleLogicalResult< NA, T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Functions

- template<bool NA, typename T >
  Rcpp::sugar::Negate_SingleLogicalResult< NA, T > operator! (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)

7.298.1 Function Documentation

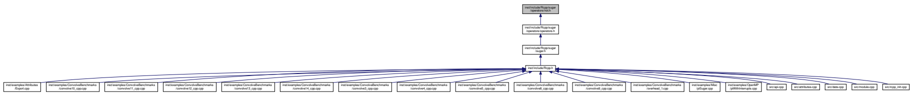
7.298.1.1 operator"!()

```
template<bool NA, typename T >
Rcpp::sugar::Negate_SingleLogicalResult<NA,T> operator!  (
             const Rcpp::sugar::SingleLogicalResult< NA, T > & x )  [inline]
```

Definition at line 62 of file not.h.

7.299 inst/include/Rcpp/sugar/operators/not.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::not_< RTYPE, NA >
- class Rcpp::sugar::not_< RTYPE, false >
- class Rcpp::sugar::not_< REALSXP, NA >
- class Rcpp::sugar::not_< REALSXP, false >
- class Rcpp::sugar::not_< CPLXSXP, NA >
- class Rcpp::sugar::not_< CPLXSXP, false >
- class Rcpp::sugar::Not_Vector< RTYPE, NA, T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Not_Vector< RTYPE, NA, T > operator! (const Rcpp::VectorBase< RTYPE, NA, T > &x)

7.299.1 Function Documentation

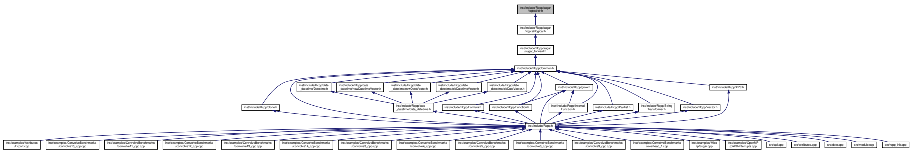
7.299.1.1 operator"!()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Not_Vector< RTYPE , NA , T > operator!  (
             const Rcpp::VectorBase< RTYPE, NA, T > & x )  [inline]
```

Definition at line 101 of file not.h.

7.300 inst/include/Rcpp/sugar/logical/or.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >
- class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Functions

- template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > operator|| (const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs, const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &rhs)
- template<bool LHS_NA, typename LHS_T >
  Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > operator|| (const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs, bool rhs)
- template<bool LHS_NA, typename LHS_T >
  Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > operator|| (bool rhs, const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs)
- template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > operator| (const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > &rhs)

7.300.1 Function Documentation

7.300.1.1 operator"|()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression<LHS_NA,LHS_T,RHS_NA,RHS_T> operator| (
            const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > & lhs,
            const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 315 of file or.h.

7.300.1.2 operator"|"|() [1/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool<LHS_NA,LHS_T> operator|| (
            bool rhs,
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs )  [inline]
```

Definition at line 305 of file or.h.

7.300.1.3 operator"|"|() [2/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool<LHS_NA,LHS_T> operator|| (
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
            bool rhs )  [inline]
```

Definition at line 296 of file or.h.

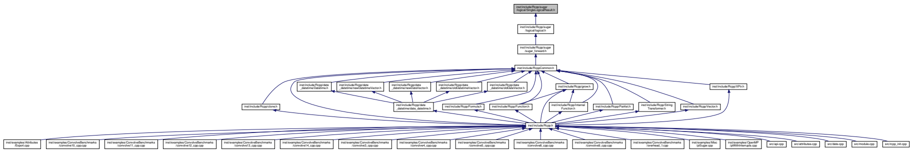
7.300.1.4 operator"|"|() [3/3]

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_T,RHS_NA,RHS_T> operator|| (
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
            const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & rhs )  [inline]
```

Definition at line 287 of file or.h.

7.301 inst/include/Rcpp/sugar/logical/SingleLogicalResult.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::forbidden_conversion< true >
- class Rcpp::sugar::conversion_to_bool_is_forbidden< x >
- class Rcpp::sugar::SingleLogicalResult< NA, T >

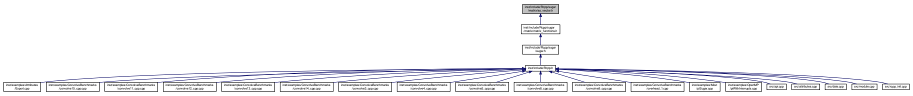
Namespaces

- Rcpp

    Rcpp API.
- Rcpp::sugar

7.302 inst/include/Rcpp/sugar/matrix/as_vector.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.
- Rcpp::internal

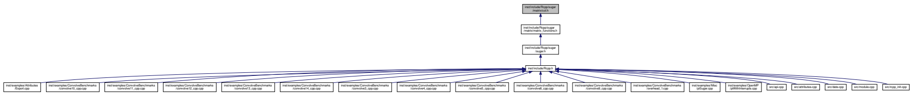
    internal implementation details

Functions

- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > Rcpp::internal::as_vector__impl (MatrixBase< RTYPE, NA, T > &t, Rcpp::traits::false_type)
- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > Rcpp::internal::as_vector__impl (MatrixBase< RTYPE, NA, T > &t, Rcpp::traits::true_type)
- template<int RTYPE, bool NA, typename T >
  Rcpp::Vector< RTYPE > Rcpp::as_vector (const MatrixBase< RTYPE, NA, T > &t)

7.303 inst/include/Rcpp/sugar/matrix/col.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/matrix/col.h
inst/include/Rcpp/sugar/matrix/matrix_functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >

Namespaces

- Rcpp

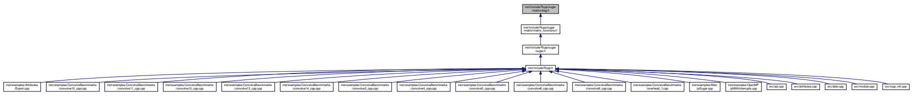
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Col< RTYPE, LHS_NA, LHS_T > Rcpp::col (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > &lhs)

7.304 inst/include/Rcpp/sugar/matrix/diag.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/matrix/diag.h
inst/include/Rcpp/sugar/matrix/matrix_functions.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >
- class Rcpp::sugar::Diag_Maker< RTYPE, NA, T >
- struct Rcpp::sugar::diag_result_type_trait< T >

Namespaces

- Rcpp

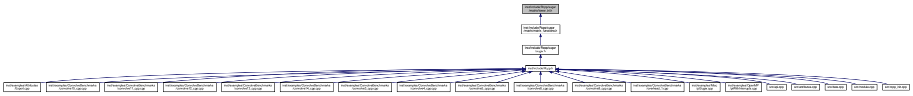
  Rcpp API.
- Rcpp::sugar

Functions

- template<typename T >
  sugar::diag_result_type_trait< T >::type Rcpp::diag (const T &t)

7.305 inst/include/Rcpp/sugar/matrix/lower_tri.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::LowerTri< RTYPE, NA, T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

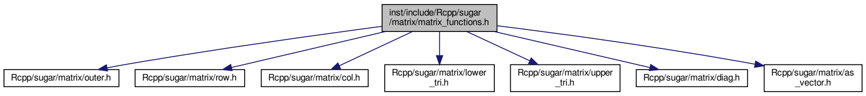
Functions

- template<int RTYPE, bool NA, typename T >
  sugar::LowerTri< RTYPE, NA, T > Rcpp::lower_tri (const Rcpp::MatrixBase< RTYPE, NA, T > &lhs, bool diag=false)

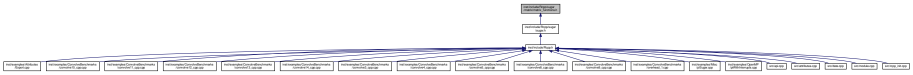
7.306 inst/include/Rcpp/sugar/matrix/matrix_functions.h File Reference

```
#include <Rcpp/sugar/matrix/outer.h>
#include <Rcpp/sugar/matrix/row.h>
#include <Rcpp/sugar/matrix/col.h>
#include <Rcpp/sugar/matrix/lower_tri.h>
#include <Rcpp/sugar/matrix/upper_tri.h>
#include <Rcpp/sugar/matrix/diag.h>
#include <Rcpp/sugar/matrix/as_vector.h>
```

Include dependency graph for matrix_functions.h:

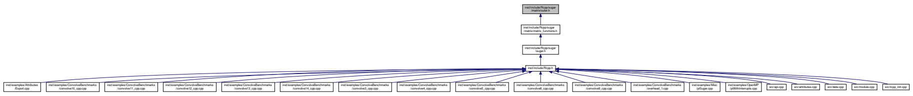

inst/include/Rcpp/sugar/matrix/matrix_functions.h
Rcpp/sugar/matrix/outer.h
Rcpp/sugar/matrix/row.h
Rcpp/sugar/matrix/col.h
Rcpp/sugar/matrix/lower_tri.h
Rcpp/sugar/matrix/upper_tri.h
Rcpp/sugar/matrix/diag.h
Rcpp/sugar/matrix/as_vector.h

This graph shows which files directly or indirectly include this file:



7.307 inst/include/Rcpp/sugar/matrix/outer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >

Namespaces

- Rcpp

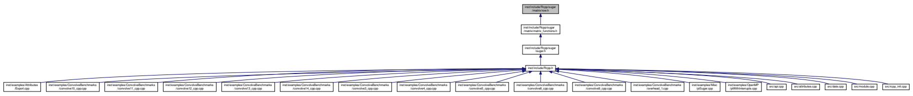
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function >
  sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > Rcpp::outer (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs, Function fun)

7.308 inst/include/Rcpp/sugar/matrix/row.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >

Namespaces

- Rcpp

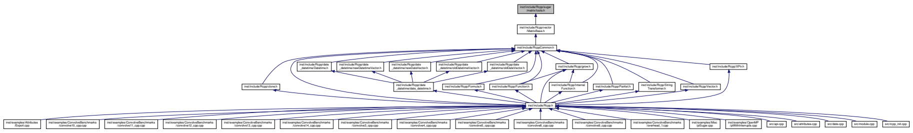
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T >
  sugar::Row< RTYPE, LHS_NA, LHS_T > Rcpp::row (const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > &lhs)

7.309 inst/include/Rcpp/sugar/matrix/tools.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/matrix/tools.h
inst/include/Rcpp/vector/MatrixBase.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

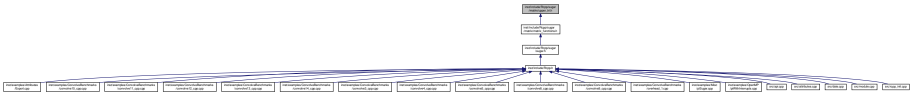
  internal implementation details

Functions

- int Rcpp::internal::get_line (int index, int nr)
- int Rcpp::internal::get_column (int index, int nr)
- int Rcpp::internal::get_column (int index, int nr, int i)

7.310 inst/include/Rcpp/sugar/matrix/upper_tri.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::UpperTri< RTYPE, NA, T >

Namespaces

- Rcpp

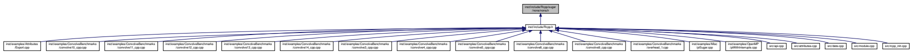
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  sugar::UpperTri< RTYPE, NA, T > Rcpp::upper_tri (const Rcpp::MatrixBase< RTYPE, NA, T > &lhs, bool diag=false)

7.311 inst/include/Rcpp/sugar/nona/nona.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Nona< RTYPE, NA, VECTOR >
- class Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >
- class Rcpp::sugar::NonaPrimitive< T >

Namespaces

- Rcpp

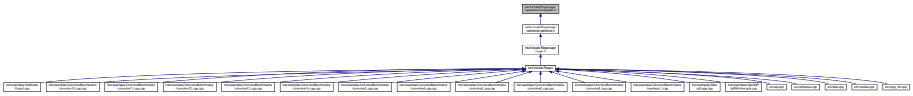
  Rcpp API.
- Rcpp::sugar

Functions

- template< int RTYPE, bool NA, typename VECTOR >
  sugar::Nona< RTYPE, NA, VECTOR > Rcpp::noNA (const Rcpp::VectorBase< RTYPE, NA, VECTOR > &vec)
- sugar::NonaPrimitive< double > Rcpp::noNA (double x)
- sugar::NonaPrimitive< int > Rcpp::noNA (int x)

7.312 inst/include/Rcpp/sugar/operators/Comparator.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >

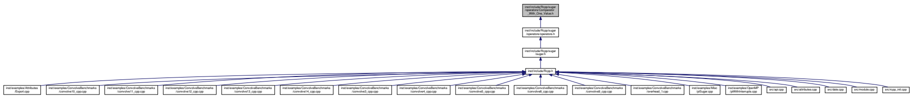
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

7.313 inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >
- class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >

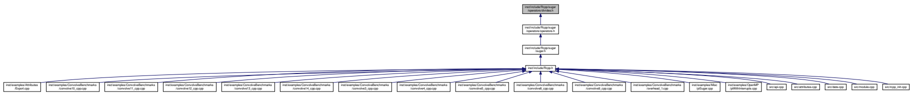
Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

7.314 inst/include/Rcpp/sugar/operators/divides.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/operators/divides.h
inst/include/Rcpp/sugar/operators/operators.h
inst/include/Rcpp/sugar/sugar.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

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

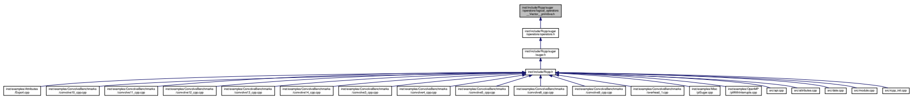
  Rcpp API.
- Rcpp::sugar

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 VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.315 inst/include/Rcpp/sugar/operators/logical_operators__Vector__primitive.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::less< RTYPE >, NA, T > operator< (const Rcpp::VectorBase< RTYPE, NA, T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::less< RTYPE >, NA, T > operator> (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< RTYPE >::type rhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::less_or_equal< RTYPE >, NA, T > operator>= (typename Rcpp::traits::storage_type< RTYPE >::type rhs, const Rcpp::VectorBase< RTYPE, NA, T > &lhs)

- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, NA, T > operator>= (const Rcpp::VectorBase< RTYPE, NA, T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, NA, T > operator<= (typename Rcpp::traits::storage_type< RTYPE >::type rhs, const Rcpp::VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::equal< RTYPE >, NA, T > operator== (const Rcpp::VectorBase< RTYPE, NA, T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::equal< RTYPE >, NA, T > operator== (typename Rcpp::traits::storage_type< RTYPE >::type rhs, const Rcpp::VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::not_equal< RTYPE >, NA, T > operator!= (const Rcpp::VectorBase< RTYPE, NA, T > &lhs, typename Rcpp::traits::storage_type< RTYPE >::type rhs)
- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::not_equal< RTYPE >, NA, T > operator!= (typename Rcpp::traits::storage_type< RTYPE >::type rhs, const Rcpp::VectorBase< RTYPE, NA, T > &lhs)

7.315.1 Function Documentation

7.315.1.1 operator"!=() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::not_equal<RTYPE>, NA, T > operator!=
(
             const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
             typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 255 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.2 operator"!=() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::not_equal<RTYPE>, NA, T > operator!=
(
             typename Rcpp::traits::storage_type< RTYPE >::type rhs,
             const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 275 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.3 operator<() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less<RTYPE>, NA, T > operator< (
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
            typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 1 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.4 operator<() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater<RTYPE>, NA, T > operator< (
            typename Rcpp::traits::storage_type< RTYPE >::type rhs,
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 78 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.5 operator<=() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, NA, T > operator<=
(
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
            typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 78 of file logical_operators__Vector__primitive.h.

7.315.1.6 operator<=() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, NA, T >
operator<= (
            typename Rcpp::traits::storage_type< RTYPE >::type rhs,
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 167 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.7 operator==() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::equal<RTYPE>, NA, T > operator== (
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
            typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 211 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.8 operator==() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::equal<RTYPE>, NA, T > operator== (
            typename Rcpp::traits::storage_type< RTYPE >::type rhs,
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 231 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.9 operator>() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater<RTYPE>, NA, T > operator> (
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
            typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 78 of file logical_operators__Vector__primitive.h.

7.315.1.10 operator>() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less<RTYPE>, NA, T > operator> (
            typename Rcpp::traits::storage_type< RTYPE >::type rhs,
            const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 54 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.315.1.11 operator>=() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, NA, T >
operator>= (
             const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
             typename Rcpp::traits::storage_type< RTYPE >::type rhs )  [inline]
```

Definition at line 167 of file logical_operators__Vector__primitive.h.

7.315.1.12 operator>=() [2/2]

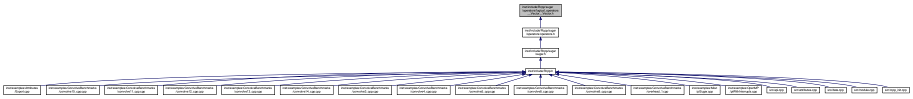
```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, NA, T > operator>=
(
             typename Rcpp::traits::storage_type< RTYPE >::type rhs,
             const Rcpp::VectorBase< RTYPE, NA, T > & lhs )  [inline]
```

Definition at line 142 of file logical_operators__Vector__primitive.h.

References Rcpp::NA.

7.316 inst/include/Rcpp/sugar/operators/logical_operators__Vector__Vector.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::less< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T > operator< (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::greater< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T > operator> (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::less_or_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T > operator<= (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T > operator>= (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T > operator== (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::not_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_↩
  T > operator!= (const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.316.1 Function Documentation

7.316.1.1 operator"!=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::not_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T >
operator!= (
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 139 of file logical_operators__Vector__Vector.h.

7.316.1.2 operator<()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::less<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T > operator<
(
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 1 of file logical_operators__Vector__Vector.h.

7.316.1.3 operator<=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T
> operator<= (
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 55 of file logical_operators__Vector__Vector.h.

7.316.1.4 operator==()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T > operator==
(
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 118 of file logical_operators__Vector__Vector.h.

7.316.1.5 operator>()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::greater<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T >
operator> (
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 55 of file logical_operators__Vector__Vector.h.

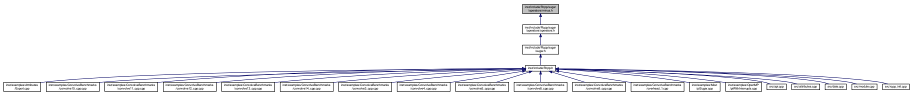
7.316.1.6 operator>=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA,
RHS_T > operator>= (
             const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
             const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs )  [inline]
```

Definition at line 97 of file logical_operators__Vector__Vector.h.

7.317 inst/include/Rcpp/sugar/operators/minus.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >
- class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >
- class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >
- class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >
- class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >
- class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >
- class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >
- class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

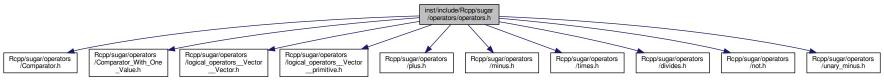
Functions

- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Vector_Primitive< RTYPE, NA, T > >::type Rcpp::operator- (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Primitive_Vector< RTYPE, NA, T > >::type Rcpp::operator- (const U &lhs, const VectorBase< RTYPE, NA, T > &rhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator- (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

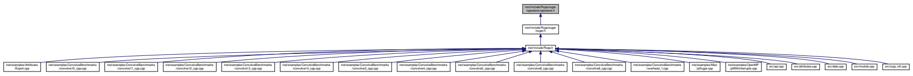
7.318 inst/include/Rcpp/sugar/operators/operators.h File Reference

```
#include <Rcpp/sugar/operators/Comparator.h>
#include <Rcpp/sugar/operators/Comparator_With_One_Value.h>
#include <Rcpp/sugar/operators/logical_operators__Vector__Vector.h>
#include <Rcpp/sugar/operators/logical_operators__Vector__primitive.h>
#include <Rcpp/sugar/operators/plus.h>
#include <Rcpp/sugar/operators/minus.h>
#include <Rcpp/sugar/operators/times.h>
#include <Rcpp/sugar/operators/divides.h>
#include <Rcpp/sugar/operators/not.h>
#include <Rcpp/sugar/operators/unary_minus.h>
```

Include dependency graph for operators.h:

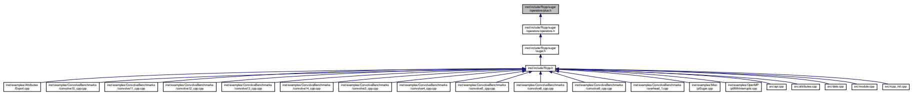


This graph shows which files directly or indirectly include this file:



7.319 inst/include/Rcpp/sugar/operators/plus.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >

- class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >
- class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >
- class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >
- class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >
- class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >
- class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >
- class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >

Namespaces

- Rcpp

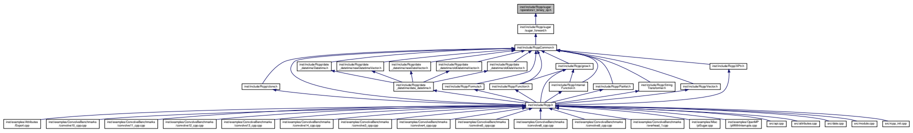
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >↩ ::type Rcpp::operator+ (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >↩ ::type Rcpp::operator+ (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type Rcpp::operator+ (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type Rcpp::operator+ (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator+ (const Vector↩ Base< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.320 inst/include/Rcpp/sugar/operators/r_binary_op.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/sugar/operators/r_binary_op.h
inst/include/Rcpp/sugar/sugar_forward.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Macros

- #define RCPP_OP(NAME, OP)

7.320.1 Macro Definition Documentation

7.320.1.1 RCPP_OP

```
#define RCPP_OP(
               NAME,
               OP )
```

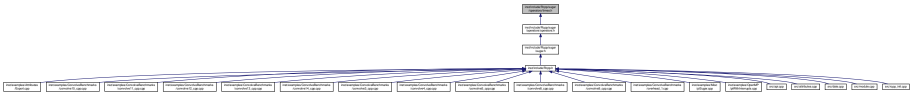
Value:

```
template <int RTYPE>                                             \
class NAME {                                                     \
public:                                                          \
        typedef typename traits::storage_type<RTYPE>::type STORAGE ;    \
        inline int operator()( STORAGE lhs, STORAGE rhs) const {       \
                return lhs OP rhs ;                                         \
        }                                                               \
} ;
```

Definition at line 29 of file r_binary_op.h.

7.321 inst/include/Rcpp/sugar/operators/times.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
- class Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >
- class Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >
- class Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >
- class Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >
- class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >
- class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >
- class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >
- class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >

Namespaces

- Rcpp

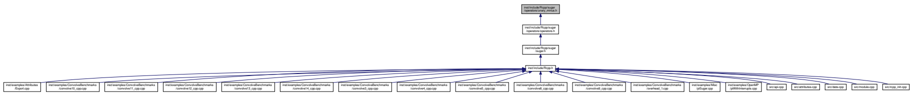
  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive< RTYPE, NA, T > >::type Rcpp::operator∗ (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive< RTYPE, NA, T > >::type Rcpp::operator∗ (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > >::type Rcpp::operator∗ (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)
- template<int RTYPE, bool NA, typename T , typename U >
  traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > >::type Rcpp::operator∗ (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
  sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator∗ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)

7.322 inst/include/Rcpp/sugar/operators/unary_minus.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::sugar::unary_minus_result_type< RTYPE >
- struct Rcpp::sugar::unary_minus_result_type< LGLSXP >
- class Rcpp::sugar::unary_minus< RTYPE, NA >
- class Rcpp::sugar::unary_minus< RTYPE, false >
- class Rcpp::sugar::unary_minus< CPLXSXP, NA >
- class Rcpp::sugar::unary_minus< CPLXSXP, false >
- class Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::sugar

Functions

- template<int RTYPE, bool NA, typename T >
  Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > operator- (const Rcpp::VectorBase< RTYPE, NA, T > &x)

7.322.1 Function Documentation

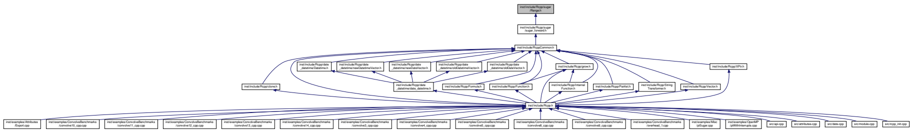
7.322.1.1 operator-()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UnaryMinus_Vector< RTYPE , NA , T > operator- (
              const Rcpp::VectorBase< RTYPE, NA, T > & x )  [inline]
```

Definition at line 113 of file unary_minus.h.

7.323 inst/include/Rcpp/sugar/Range.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Range

Namespaces

- Rcpp

  Rcpp API.

7.324 inst/include/Rcpp/sugar/sets.h File Reference

Macros

- #define RCPP_UNORDERED_SET std::set
- #define RCPP_UNORDERED_MAP std::map

7.324.1 Macro Definition Documentation

7.324.1.1 RCPP_UNORDERED_MAP

```
#define RCPP_UNORDERED_MAP std::map
```

Definition at line 58 of file sets.h.

7.324.1.2 RCPP_UNORDERED_SET

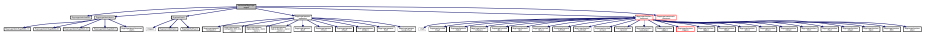
```
#define RCPP_UNORDERED_SET std::set
```

Definition at line 57 of file sets.h.

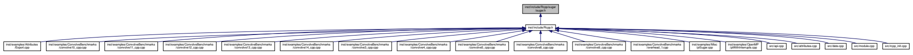
7.325 inst/include/Rcpp/sugar/sugar.h File Reference

```
#include <Rcpp/sugar/tools/iterator.h>
#include <Rcpp/sugar/block/block.h>
#include <Rcpp/hash/hash.h>
#include <Rcpp/sugar/operators/operators.h>
#include <Rcpp/sugar/functions/functions.h>
#include <Rcpp/sugar/matrix/matrix_functions.h>
```

Include dependency graph for sugar.h:



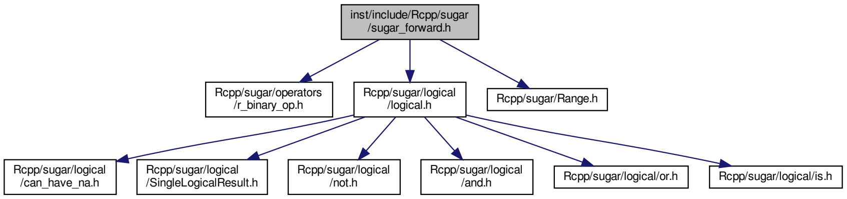
This graph shows which files directly or indirectly include this file:



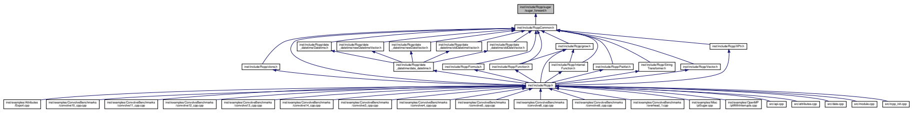
7.326 inst/include/Rcpp/sugar/sugar_forward.h File Reference

```
#include <Rcpp/sugar/operators/r_binary_op.h>
#include <Rcpp/sugar/logical/logical.h>
#include <Rcpp/sugar/Range.h>
```

Include dependency graph for sugar_forward.h:

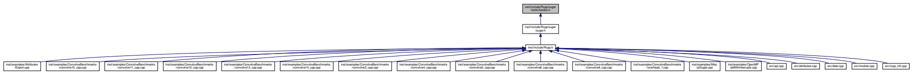


This graph shows which files directly or indirectly include this file:



7.327 inst/include/Rcpp/sugar/tools/iterator.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::sugar::SugarIterator< T >
- struct Rcpp::sugar::sugar_const_iterator_type< T >
- struct Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > >
- struct Rcpp::sugar::sugar_const_iterator_type< CharacterVector >
- struct Rcpp::sugar::is_sugar_vector< T >
- struct Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >

Namespaces

- Rcpp

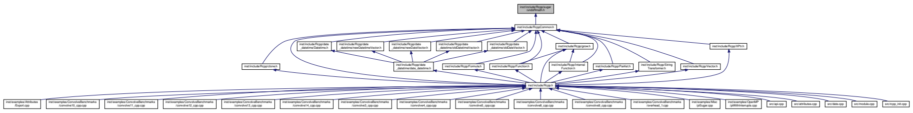
  Rcpp API.
- Rcpp::sugar

Functions

- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_begin__impl (const T &obj, Rcpp::traits::true_type)
- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_begin__impl (const T &obj, Rcpp::traits::false_type)
- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_begin (const T &obj)
- template<> SEXP ∗ Rcpp::sugar::get_const_begin (const CharacterVector &obj)
- template<typename T >
  sugar_const_iterator_type< T >::type Rcpp::sugar::get_const_end (const T &obj)

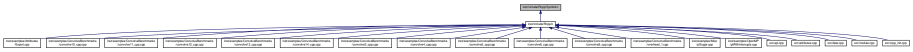
7.328 inst/include/Rcpp/sugar/undoRmath.h File Reference

This graph shows which files directly or indirectly include this file:



7.329 inst/include/Rcpp/Symbol.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

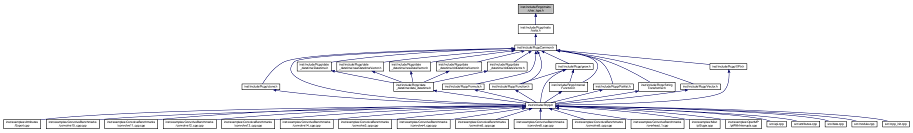
- typedef Symbol_Impl< NoProtectStorage > Rcpp::Symbol

Functions

- Rcpp::RCPP_API_CLASS (Symbol_Impl)

7.330 inst/include/Rcpp/traits/char_type.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::char_type< T >
- struct Rcpp::traits::char_type< const wchar_t ∗ >
- struct Rcpp::traits::char_type< const char ∗ >

Namespaces

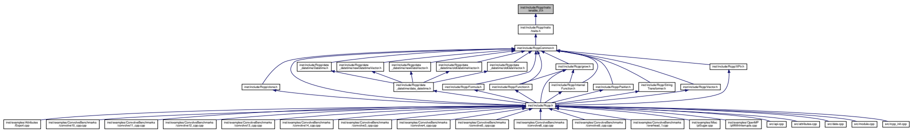
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.331 inst/include/Rcpp/traits/enable_if.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/enable_if.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::traits::enable_if< B, T >
- struct Rcpp::traits::enable_if< true, T >

Namespaces

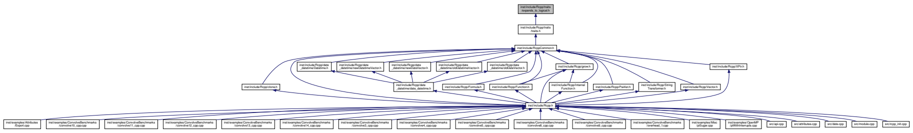
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.332 inst/include/Rcpp/traits/expands_to_logical.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/expands_to_logical.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

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

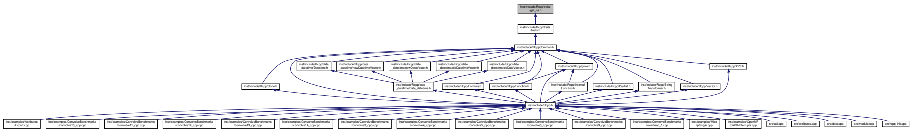
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.333 inst/include/Rcpp/traits/get_na.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

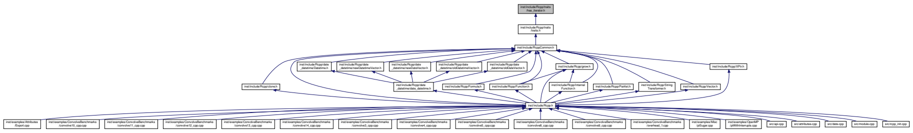
  traits used to dispatch wrap

Functions

- template<int RTYPE>
  storage_type< RTYPE >::type Rcpp::traits::get_na ()
- template<> int Rcpp::traits::get_na< INTSXP > ()
- template<> int Rcpp::traits::get_na< LGLSXP > ()
- template<> double Rcpp::traits::get_na< REALSXP > ()
- template<> Rcomplex Rcpp::traits::get_na< CPLXSXP > ()
- template<> SEXP Rcpp::traits::get_na< STRSXP > ()
- template<> SEXP Rcpp::traits::get_na< VECSXP > ()

7.334 inst/include/Rcpp/traits/has_iterator.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::__sfinae_types
- struct Rcpp::traits::__sfinae_types::__two
- class Rcpp::traits::_has_iterator_helper< T >
- struct Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U >
- class Rcpp::traits::_is_importer_helper< T >
- struct Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U >
- class Rcpp::traits::_is_generator_helper< T >
- struct Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U >
- class Rcpp::traits::_is_exporter_helper< T >
- struct Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U >
- struct Rcpp::traits::has_iterator< T >
- struct Rcpp::traits::is_importer< T >
- struct Rcpp::traits::is_exporter< T >
- struct Rcpp::traits::is_generator< T >

Namespaces

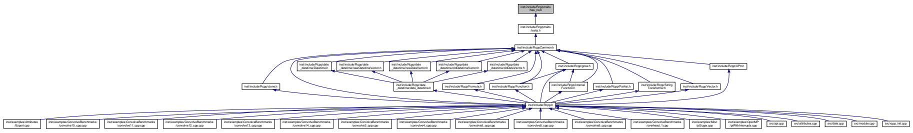
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.335 inst/include/Rcpp/traits/has_na.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::has_na< RTYPE >
- struct Rcpp::traits::has_na< INTSXP >
- struct Rcpp::traits::has_na< REALSXP >
- struct Rcpp::traits::has_na< CPLXSXP >
- struct Rcpp::traits::has_na< STRSXP >
- struct Rcpp::traits::has_na< LGLSXP >

Namespaces

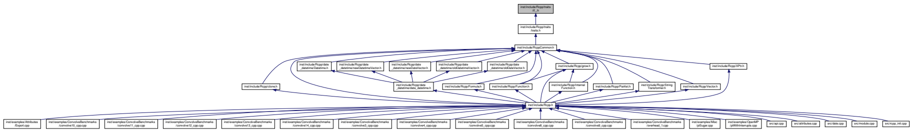
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.336 inst/include/Rcpp/traits/if_.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::if_< COND, LHS, RHS >
- struct Rcpp::traits::if_< false, LHS, RHS >

Namespaces

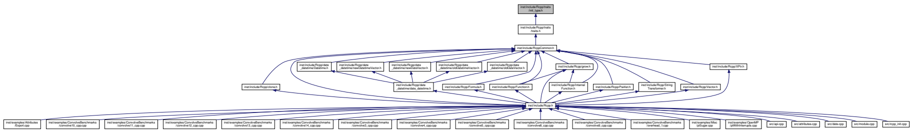
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.337 inst/include/Rcpp/traits/init_type.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::init_type< RTYPE >
- struct Rcpp::traits::init_type< STRSXP >
- struct Rcpp::traits::init_type< LGLSXP >

Namespaces

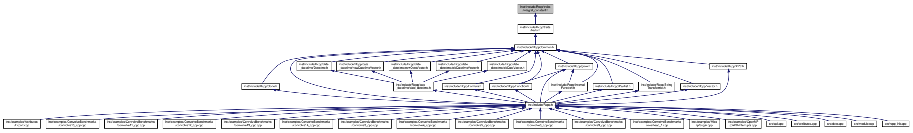
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.338 inst/include/Rcpp/traits/integral_constant.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::integral_constant< _T, _V >
- struct Rcpp::traits::both< T, U >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

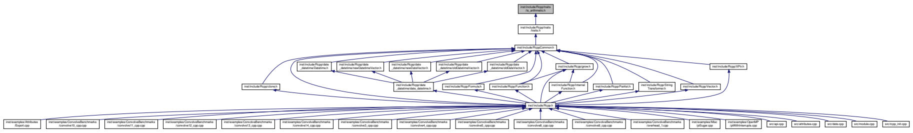
  traits used to dispatch wrap

Typedefs

- typedef integral_constant< bool, true > Rcpp::traits::true_type
- typedef integral_constant< bool, false > Rcpp::traits::false_type

7.339 inst/include/Rcpp/traits/is_arithmetic.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_arithmetic< typename >
- struct Rcpp::traits::is_arithmetic< short >
- struct Rcpp::traits::is_arithmetic< const short >
- struct Rcpp::traits::is_arithmetic< unsigned short >
- struct Rcpp::traits::is_arithmetic< const unsigned short >
- struct Rcpp::traits::is_arithmetic< int >
- struct Rcpp::traits::is_arithmetic< const int >
- struct Rcpp::traits::is_arithmetic< unsigned int >
- struct Rcpp::traits::is_arithmetic< const unsigned int >
- struct Rcpp::traits::is_arithmetic< long >
- struct Rcpp::traits::is_arithmetic< const long >
- struct Rcpp::traits::is_arithmetic< unsigned long >
- struct Rcpp::traits::is_arithmetic< const unsigned long >
- struct Rcpp::traits::is_arithmetic< float >
- struct Rcpp::traits::is_arithmetic< const float >
- struct Rcpp::traits::is_arithmetic< double >
- struct Rcpp::traits::is_arithmetic< const double >
- struct Rcpp::traits::is_arithmetic< long double >
- struct Rcpp::traits::is_arithmetic< const long double >

Namespaces

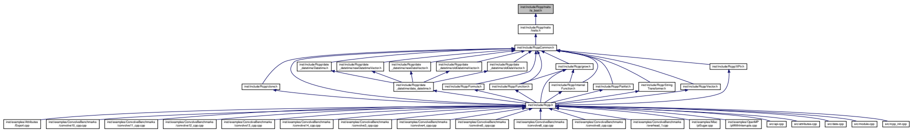
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.340 inst/include/Rcpp/traits/is_bool.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_bool< typename >
- struct Rcpp::traits::is_bool< bool >
- struct Rcpp::traits::is_bool< const bool >
- struct Rcpp::traits::is_bool< volatile bool >

Namespaces

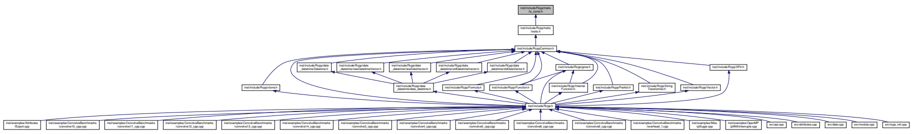
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.341 inst/include/Rcpp/traits/is_const.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_const< typename >

  type properties [4.5.3].
- struct Rcpp::traits::is_const< _Tp const >

Namespaces

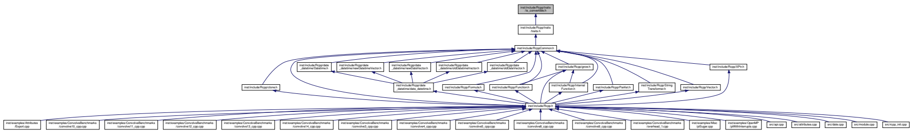
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.342 inst/include/Rcpp/traits/is_convertible.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::traits::is_convertible< T, U >
- struct Rcpp::traits::is_convertible< T, U >::Big

Namespaces

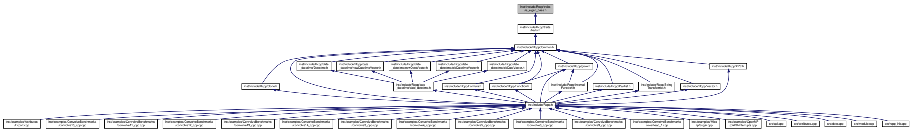
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.343 inst/include/Rcpp/traits/is_eigen_base.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::traits::_is_eigen_helper< T >
- struct Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U >
- struct Rcpp::traits::is_eigen_base< T >

Namespaces

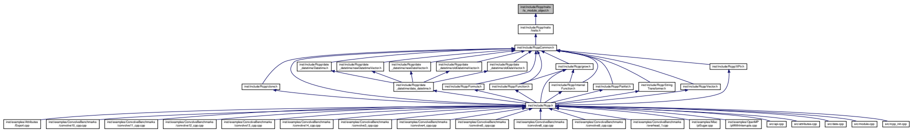
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.344 inst/include/Rcpp/traits/is_module_object.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_module_object< T >

Namespaces

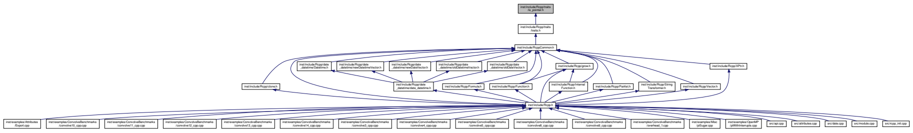
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.345 inst/include/Rcpp/traits/is_pointer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_pointer< T >
- struct Rcpp::traits::is_pointer< T ∗ >

Namespaces

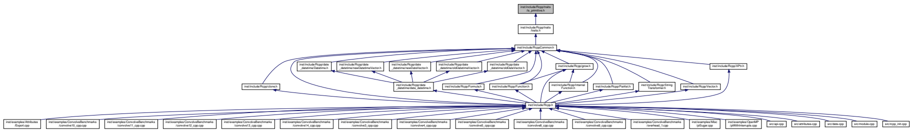
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.346 inst/include/Rcpp/traits/is_primitive.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_primitive< T >

Namespaces

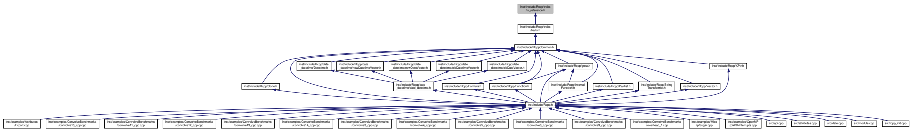
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.347 inst/include/Rcpp/traits/is_reference.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::is_reference< typename >
- struct Rcpp::traits::is_reference< _Tp & >

Namespaces

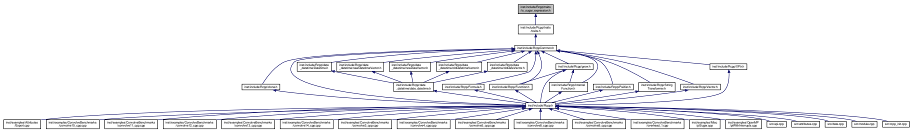
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.348 inst/include/Rcpp/traits/is_sugar_expression.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::traits::_is_sugar_expression_helper< T >
- struct Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U >
- struct Rcpp::traits::is_sugar_expression< T >

Namespaces

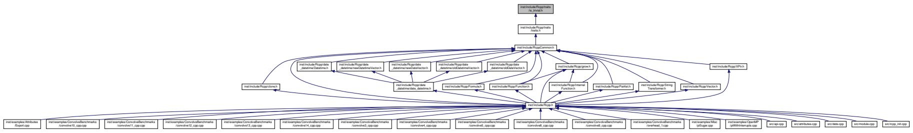
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.349 inst/include/Rcpp/traits/is_trivial.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/is_trivial.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::traits::is_trivial< RTYPE >
- struct Rcpp::traits::is_trivial< VECSXP >
- struct Rcpp::traits::is_trivial< EXPRSXP >

Namespaces

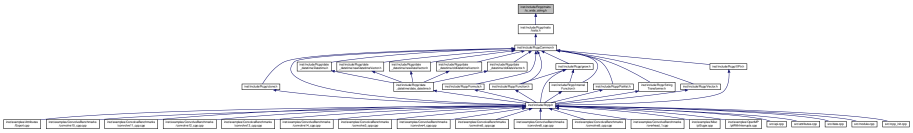
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.350 inst/include/Rcpp/traits/is_wide_string.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/is_wide_string.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::traits::is_wide_string< T >
- struct Rcpp::traits::is_wide_string< const wchar_t ∗ >
- struct Rcpp::traits::is_wide_string< const char ∗ >
- struct Rcpp::traits::is_wide_string< wchar_t >
- struct Rcpp::traits::is_wide_string< char >

Namespaces

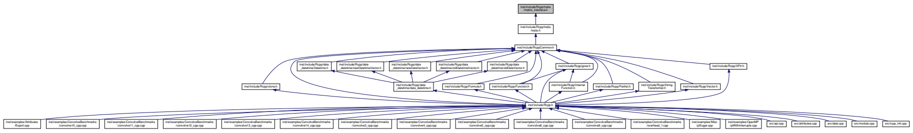
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.351 inst/include/Rcpp/traits/matrix_interface.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::traits::_has_matrix_interface_helper< T >
- struct Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U >
- struct Rcpp::traits::matrix_interface< T >

Namespaces

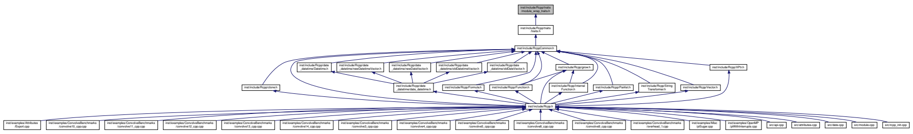
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.352 inst/include/Rcpp/traits/module_wrap_traits.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::normal_wrap_tag
- struct Rcpp::traits::void_wrap_tag
- struct Rcpp::traits::pointer_wrap_tag
- struct Rcpp::traits::module_wrap_traits< T >
- struct Rcpp::traits::module_wrap_traits< void >
- struct Rcpp::traits::module_wrap_traits< T ∗ >

Namespaces

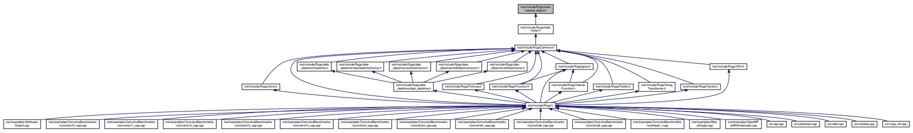
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.353 inst/include/Rcpp/traits/named_object.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::needs_protection< T >
- struct Rcpp::traits::needs_protection< SEXP >
- class Rcpp::traits::named_object< T >
- class Rcpp::traits::named_object< SEXP >
- struct Rcpp::traits::is_named< T >
- struct Rcpp::traits::is_named< named_object< T > >
- struct Rcpp::traits::is_named< Rcpp::Argument >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.354 inst/include/Rcpp/traits/num2type.h File Reference

Classes

- struct Rcpp::traits::num2type< N >

Namespaces

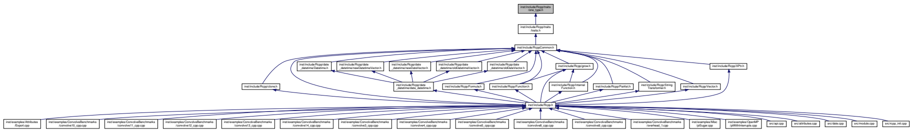
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.355 inst/include/Rcpp/traits/one_type.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::allowed_matrix_type< true >
- class Rcpp::traits::one_type< T >
- class Rcpp::traits::zero_type< T >

Namespaces

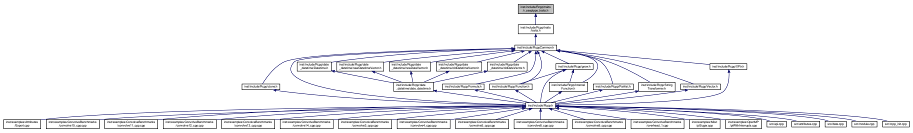
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.356 inst/include/Rcpp/traits/r_sexptype_traits.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::r_sexptype_traits< T >
- struct Rcpp::traits::r_sexptype_traits< int >
- struct Rcpp::traits::r_sexptype_traits< const int >
- struct Rcpp::traits::r_sexptype_traits< double >
- struct Rcpp::traits::r_sexptype_traits< const double >
- struct Rcpp::traits::r_sexptype_traits< bool >
- struct Rcpp::traits::r_sexptype_traits< std::string >
- struct Rcpp::traits::r_sexptype_traits< Rcomplex >
- struct Rcpp::traits::r_sexptype_traits< Rbyte >
- struct Rcpp::traits::r_sexptype_traits< unsigned int >
- struct Rcpp::traits::r_sexptype_traits< float >
- struct Rcpp::traits::r_sexptype_traits< long >
- struct Rcpp::traits::r_sexptype_traits< unsigned long >
- struct Rcpp::traits::r_sexptype_traits< long double >
- struct Rcpp::traits::r_sexptype_traits< short >
- struct Rcpp::traits::r_sexptype_traits< unsigned short >
- struct Rcpp::traits::r_sexptype_traits< std::complex< double > >
- struct Rcpp::traits::r_sexptype_traits< std::complex< float > >
- struct Rcpp::traits::r_sexptype_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

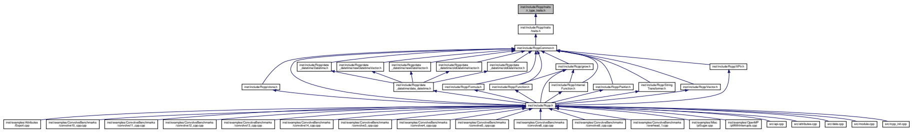
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.357 inst/include/Rcpp/traits/r_type_traits.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/r_type_traits.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/InternalFunction.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/StringTransformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::traits::r_type_primitive_tag
- struct Rcpp::traits::r_type_string_tag
- struct Rcpp::traits::r_type_generic_tag
- struct Rcpp::traits::r_type_pairstring_primitive_tag
- struct Rcpp::traits::r_type_pairstring_string_tag
- struct Rcpp::traits::r_type_pairstring_generic_tag
- struct Rcpp::traits::r_type_pair_tag
- struct Rcpp::traits::r_type_module_object_pointer_tag
- struct Rcpp::traits::r_type_module_object_const_pointer_tag
- struct Rcpp::traits::r_type_module_object_tag
- struct Rcpp::traits::r_type_module_object_reference_tag
- struct Rcpp::traits::r_type_module_object_const_reference_tag
- struct Rcpp::traits::r_type_enum_tag
- struct Rcpp::traits::r_type_RcppString_tag
- struct Rcpp::traits::r_type_traits< T >
- struct Rcpp::traits::r_type_traits< Rcpp::object< T > >
- struct Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, T > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, int > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, const int > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, double > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, bool > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, char > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, wchar_t > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned int > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, float > >
- struct Rcpp::traits::r_type_traits< int >
- struct Rcpp::traits::r_type_traits< const int >
- struct Rcpp::traits::r_type_traits< double >
- struct Rcpp::traits::r_type_traits< const double >
- struct Rcpp::traits::r_type_traits< Rbyte >
- struct Rcpp::traits::r_type_traits< Rcomplex >
- struct Rcpp::traits::r_type_traits< bool >
- struct Rcpp::traits::r_type_traits< std::string >
- struct Rcpp::traits::r_type_traits< std::wstring >
- struct Rcpp::traits::r_type_traits< char >
- struct Rcpp::traits::r_type_traits< wchar_t >
- struct Rcpp::traits::r_type_traits< unsigned int >
- struct Rcpp::traits::r_type_traits< float >
- struct Rcpp::traits::r_type_traits< const char * >
- struct Rcpp::traits::r_type_traits< const wchar_t * >
- struct Rcpp::traits::r_type_traits< long >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, long > >
- struct Rcpp::traits::r_type_traits< unsigned long >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned long > >

- struct Rcpp::traits::r_type_traits< long double >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, long double > >
- struct Rcpp::traits::r_type_traits< short >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, short > >
- struct Rcpp::traits::r_type_traits< unsigned short >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned short > >
- struct Rcpp::traits::r_type_traits< std::complex< double > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > >
- struct Rcpp::traits::r_type_traits< std::complex< float > >
- struct Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > >

Namespaces

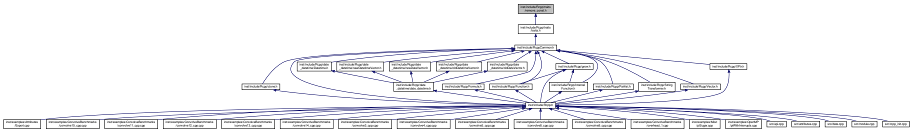
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.358 inst/include/Rcpp/traits/remove_const.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/traits/remove_const.h
inst/include/Rcpp/traits/traits.h
inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- struct Rcpp::traits::remove_const< _Tp >

  const-volatile modifications [4.7.1].
- struct Rcpp::traits::remove_const< _Tp const >

Namespaces

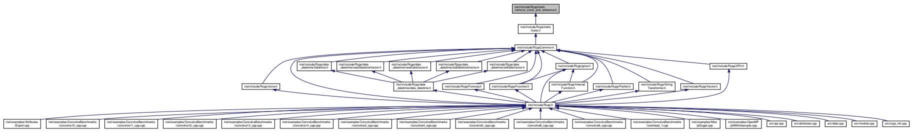
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.359 inst/include/Rcpp/traits/remove_const_and_reference.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::remove_const_and_reference< T >

Namespaces

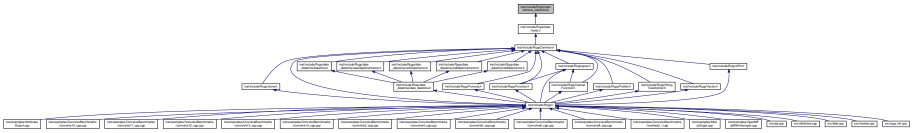
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.360 inst/include/Rcpp/traits/remove_reference.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::remove_reference< _Tp >

  reference modifications [4.7.2].
- struct Rcpp::traits::remove_reference< _Tp & >

Namespaces

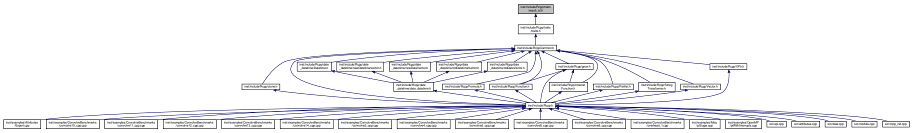
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.361 inst/include/Rcpp/traits/result_of.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::result_of< T >
- struct Rcpp::traits::result_of< RESULT_TYPE(∗)(INPUT_TYPE) >
- struct Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2) >
- struct Rcpp::traits::result_of< RESULT_TYPE(∗)(U1, U2, U3) >

Namespaces

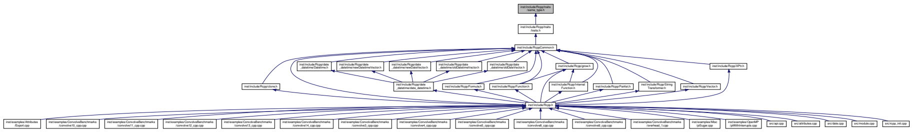
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.362 inst/include/Rcpp/traits/same_type.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::same_type< T, U >
- struct Rcpp::traits::same_type< T, T >

Namespaces

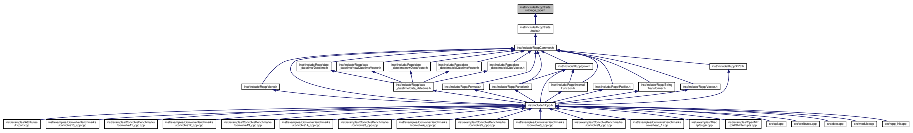
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.363 inst/include/Rcpp/traits/storage_type.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::storage_type< RTYPE >
- struct Rcpp::traits::storage_type< INTSXP >
- struct Rcpp::traits::storage_type< REALSXP >
- struct Rcpp::traits::storage_type< CPLXSXP >
- struct Rcpp::traits::storage_type< RAWSXP >
- struct Rcpp::traits::storage_type< LGLSXP >

Namespaces

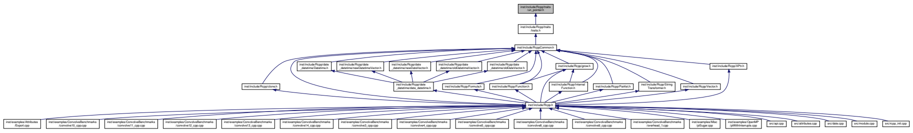
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.364 inst/include/Rcpp/traits/un_pointer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::un_pointer< T >
- struct Rcpp::traits::un_pointer< T * >
- struct Rcpp::traits::un_pointer< object< T > >

Namespaces

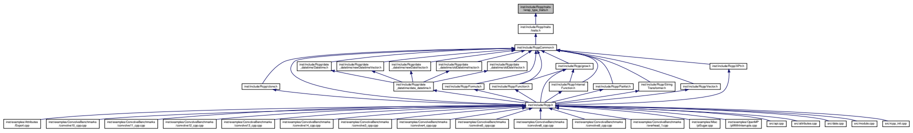
- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap

7.365 inst/include/Rcpp/traits/wrap_type_traits.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct Rcpp::traits::wrap_type_primitive_tag
- struct Rcpp::traits::wrap_type_unknown_tag
- struct Rcpp::traits::wrap_type_module_object_pointer_tag
- struct Rcpp::traits::wrap_type_module_object_tag
- struct Rcpp::traits::wrap_type_enum_tag
- struct Rcpp::traits::wrap_type_char_array
- struct Rcpp::traits::wrap_type_traits< T >
- struct Rcpp::traits::wrap_type_traits< char[N]>
- struct Rcpp::traits::wrap_type_traits< const char[N]>
- struct Rcpp::traits::wrap_type_traits< int >
- struct Rcpp::traits::wrap_type_traits< const int >
- struct Rcpp::traits::wrap_type_traits< double >
- struct Rcpp::traits::wrap_type_traits< Rbyte >
- struct Rcpp::traits::wrap_type_traits< Rcomplex >
- struct Rcpp::traits::wrap_type_traits< unsigned int >
- struct Rcpp::traits::wrap_type_traits< bool >
- struct Rcpp::traits::wrap_type_traits< std::string >
- struct Rcpp::traits::wrap_type_traits< std::wstring >
- struct Rcpp::traits::wrap_type_traits< Rcpp::String >
- struct Rcpp::traits::wrap_type_traits< char >
- struct Rcpp::traits::wrap_type_traits< wchar_t >
- struct Rcpp::traits::wrap_type_traits< float >
- struct Rcpp::traits::wrap_type_traits< std::complex< float > >
- struct Rcpp::traits::wrap_type_traits< std::complex< double > >
- struct Rcpp::traits::wrap_type_traits< long >
- struct Rcpp::traits::wrap_type_traits< unsigned long >
- struct Rcpp::traits::wrap_type_traits< long double >
- struct Rcpp::traits::wrap_type_traits< short >
- struct Rcpp::traits::wrap_type_traits< unsigned short >
- struct Rcpp::traits::wrap_type_traits< Rcpp::object< T > >

Namespaces

- Rcpp

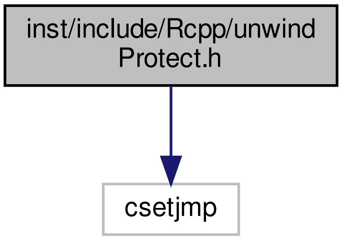
    Rcpp API.
- Rcpp::traits

    traits used to dispatch wrap

7.366 inst/include/Rcpp/unwindProtect.h File Reference

```
#include <csetjmp>
```

Include dependency graph for unwindProtect.h:



Classes

- struct Rcpp::internal::UnwindData

Namespaces

- Rcpp

    Rcpp API.
- Rcpp::internal

    internal implementation details

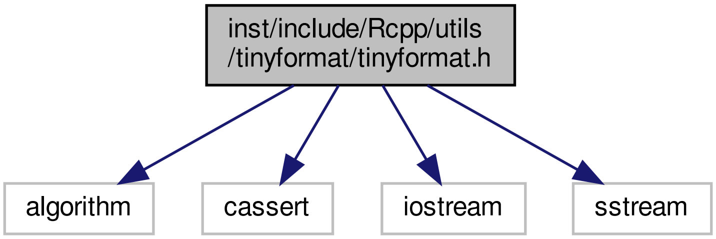
Functions

- void Rcpp::internal::maybeJump (void ∗unwind_data, Rboolean jump)
- SEXP Rcpp::unwindProtect (SEXP(∗callback)(void ∗data), void ∗data)

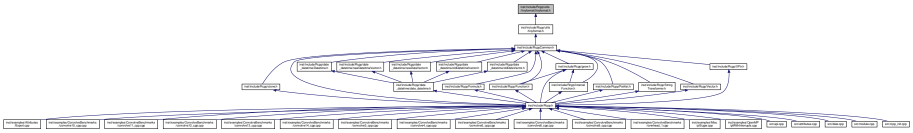
7.367 inst/include/Rcpp/utils/tinyformat/tinyformat.h File Reference

```
#include <algorithm>
#include <cassert>
#include <iostream>
#include <sstream>
```

Include dependency graph for tinyformat.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct tinyformat::detail::is_convertible< T1, T2 >
- struct tinyformat::detail::is_convertible< T1, T2 >::fail
- struct tinyformat::detail::is_convertible< T1, T2 >::succeed
- struct tinyformat::detail::is_wchar< T >
- struct tinyformat::detail::is_wchar< wchar_t ∗ >
- struct tinyformat::detail::is_wchar< const wchar_t ∗ >
- struct tinyformat::detail::is_wchar< const wchar_t[n]>
- struct tinyformat::detail::is_wchar< wchar_t[n]>
- struct tinyformat::detail::formatValueAsType< T, fmtT, convertible >
- struct tinyformat::detail::formatValueAsType< T, fmtT, true >
- struct tinyformat::detail::convertToInt< T, convertible >
- struct tinyformat::detail::convertToInt< T, true >
- class tinyformat::detail::FormatArg
- class tinyformat::FormatList
- class tinyformat::detail::FormatListN< N >
- class tinyformat::detail::FormatListN< 0 >

Namespaces

- tinyformat
- tinyformat::detail

Macros

- #define TINYFORMAT_ASSERT(cond) assert(cond)
- #define TINYFORMAT_ERROR(reason) assert(0 && reason)
- #define TINYFORMAT_HIDDEN
- #define TINYFORMAT_DEFINE_FORMAT_TRUNCATED_CSTR(type)
- #define TINYFORMAT_DEFINE_FORMATVALUE_CHAR(charType)
- #define TINYFORMAT_ARGTYPES(n) TINYFORMAT_ARGTYPES_ ## n
- #define TINYFORMAT_VARARGS(n) TINYFORMAT_VARARGS_ ## n
- #define TINYFORMAT_PASSARGS(n) TINYFORMAT_PASSARGS_ ## n
- #define TINYFORMAT_PASSARGS_TAIL(n) TINYFORMAT_PASSARGS_TAIL_ ## n
- #define TINYFORMAT_ARGTYPES_1 class T1
- #define TINYFORMAT_ARGTYPES_2 class T1, class T2
- #define TINYFORMAT_ARGTYPES_3 class T1, class T2, class T3
- #define TINYFORMAT_ARGTYPES_4 class T1, class T2, class T3, class T4
- #define TINYFORMAT_ARGTYPES_5 class T1, class T2, class T3, class T4, class T5
- #define TINYFORMAT_ARGTYPES_6 class T1, class T2, class T3, class T4, class T5, class T6
- #define TINYFORMAT_ARGTYPES_7 class T1, class T2, class T3, class T4, class T5, class T6, class T7
- #define TINYFORMAT_ARGTYPES_8 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8
- #define TINYFORMAT_ARGTYPES_9 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9
- #define TINYFORMAT_ARGTYPES_10 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10
- #define TINYFORMAT_ARGTYPES_11 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11
- #define TINYFORMAT_ARGTYPES_12 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12
- #define TINYFORMAT_ARGTYPES_13 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13
- #define TINYFORMAT_ARGTYPES_14 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14
- #define TINYFORMAT_ARGTYPES_15 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15
- #define TINYFORMAT_ARGTYPES_16 class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15, class T16
- #define TINYFORMAT_VARARGS_1 const T1& v1
- #define TINYFORMAT_VARARGS_2 const T1& v1, const T2& v2
- #define TINYFORMAT_VARARGS_3 const T1& v1, const T2& v2, const T3& v3
- #define TINYFORMAT_VARARGS_4 const T1& v1, const T2& v2, const T3& v3, const T4& v4
- #define TINYFORMAT_VARARGS_5 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5
- #define TINYFORMAT_VARARGS_6 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6
- #define TINYFORMAT_VARARGS_7 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7

- #define TINYFORMAT_VARARGS_8 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8
- #define TINYFORMAT_VARARGS_9 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9
- #define TINYFORMAT_VARARGS_10 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10
- #define TINYFORMAT_VARARGS_11 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11
- #define TINYFORMAT_VARARGS_12 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12
- #define TINYFORMAT_VARARGS_13 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13
- #define TINYFORMAT_VARARGS_14 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14
- #define TINYFORMAT_VARARGS_15 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15
- #define TINYFORMAT_VARARGS_16 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15, const T16& v16
- #define TINYFORMAT_PASSARGS_1 v1
- #define TINYFORMAT_PASSARGS_2 v1, v2
- #define TINYFORMAT_PASSARGS_3 v1, v2, v3
- #define TINYFORMAT_PASSARGS_4 v1, v2, v3, v4
- #define TINYFORMAT_PASSARGS_5 v1, v2, v3, v4, v5
- #define TINYFORMAT_PASSARGS_6 v1, v2, v3, v4, v5, v6
- #define TINYFORMAT_PASSARGS_7 v1, v2, v3, v4, v5, v6, v7
- #define TINYFORMAT_PASSARGS_8 v1, v2, v3, v4, v5, v6, v7, v8
- #define TINYFORMAT_PASSARGS_9 v1, v2, v3, v4, v5, v6, v7, v8, v9
- #define TINYFORMAT_PASSARGS_10 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
- #define TINYFORMAT_PASSARGS_11 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
- #define TINYFORMAT_PASSARGS_12 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
- #define TINYFORMAT_PASSARGS_13 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
- #define TINYFORMAT_PASSARGS_14 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
- #define TINYFORMAT_PASSARGS_15 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
- #define TINYFORMAT_PASSARGS_16 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
- #define TINYFORMAT_PASSARGS_TAIL_1
- #define TINYFORMAT_PASSARGS_TAIL_2 , v2
- #define TINYFORMAT_PASSARGS_TAIL_3 , v2, v3
- #define TINYFORMAT_PASSARGS_TAIL_4 , v2, v3, v4
- #define TINYFORMAT_PASSARGS_TAIL_5 , v2, v3, v4, v5
- #define TINYFORMAT_PASSARGS_TAIL_6 , v2, v3, v4, v5, v6
- #define TINYFORMAT_PASSARGS_TAIL_7 , v2, v3, v4, v5, v6, v7
- #define TINYFORMAT_PASSARGS_TAIL_8 , v2, v3, v4, v5, v6, v7, v8
- #define TINYFORMAT_PASSARGS_TAIL_9 , v2, v3, v4, v5, v6, v7, v8, v9
- #define TINYFORMAT_PASSARGS_TAIL_10 , v2, v3, v4, v5, v6, v7, v8, v9, v10
- #define TINYFORMAT_PASSARGS_TAIL_11 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
- #define TINYFORMAT_PASSARGS_TAIL_12 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
- #define TINYFORMAT_PASSARGS_TAIL_13 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13

- #define TINYFORMAT_PASSARGS_TAIL_14 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
- #define TINYFORMAT_PASSARGS_TAIL_15 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
- #define TINYFORMAT_PASSARGS_TAIL_16 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
- #define TINYFORMAT_FOREACH_ARGNUM(m) m(1) m(2) m(3) m(4) m(5) m(6) m(7) m(8) m(9) m(10) m(11) m(12) m(13) m(14) m(15) m(16)
- #define TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR(n)
- #define TINYFORMAT_MAKE_MAKEFORMATLIST(n)
- #define TINYFORMAT_MAKE_FORMAT_FUNCS(n)

Typedefs

- typedef const FormatList & tinyformat::FormatListRef

  Reference to type-opaque format list for passing to vformat()

Functions

- template<typename T >
  void tinyformat::detail::formatTruncated (std::ostream &out, const T &value, int ntrunc)
- template<typename T >
  void tinyformat::formatValue (std::ostream &out, const char ∗, const char ∗fmtEnd, int ntrunc, const T &value)
- int tinyformat::detail::parseIntAndAdvance (const char ∗&c)
- const char ∗ tinyformat::detail::printFormatStringLiteral (std::ostream &out, const char ∗fmt)
- const char ∗ tinyformat::detail::streamStateFromFormat (std::ostream &out, bool &spacePadPositive, int &ntrunc, const char ∗fmtStart, const detail::FormatArg ∗formatters, int &argIndex, int numFormatters)
- void tinyformat::detail::formatImpl (std::ostream &out, const char ∗fmt, const detail::FormatArg ∗formatters, int numFormatters)
- detail::FormatListN< 0 > tinyformat::makeFormatList ()
- void tinyformat::vformat (std::ostream &out, const char ∗fmt, FormatListRef list)
- void tinyformat::format (std::ostream &out, const char ∗fmt)
- std::string tinyformat::format (const char ∗fmt)
- void tinyformat::printf (const char ∗fmt)
- void tinyformat::printfln (const char ∗fmt)

7.367.1 Macro Definition Documentation

7.367.1.1 TINYFORMAT_ARGTYPES

```
#define TINYFORMAT_ARGTYPES(
              n ) TINYFORMAT_ARGTYPES_ ## n
```

Definition at line 371 of file tinyformat.h.

7.367.1.2 TINYFORMAT_ARGTYPES_1

```
#define TINYFORMAT_ARGTYPES_1 class T1
```

Definition at line 412 of file tinyformat.h.

7.367.1.3 TINYFORMAT_ARGTYPES_10

```
#define TINYFORMAT_ARGTYPES_10 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10
```

Definition at line 421 of file tinyformat.h.

7.367.1.4 TINYFORMAT_ARGTYPES_11

```
#define TINYFORMAT_ARGTYPES_11 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11
```

Definition at line 422 of file tinyformat.h.

7.367.1.5 TINYFORMAT_ARGTYPES_12

```
#define TINYFORMAT_ARGTYPES_12 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11, class T12
```

Definition at line 423 of file tinyformat.h.

7.367.1.6 TINYFORMAT_ARGTYPES_13

```
#define TINYFORMAT_ARGTYPES_13 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11, class T12, class T13
```

Definition at line 424 of file tinyformat.h.

7.367.1.7 TINYFORMAT_ARGTYPES_14

```
#define TINYFORMAT_ARGTYPES_14 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14
```

Definition at line 425 of file tinyformat.h.

7.367.1.8 TINYFORMAT_ARGTYPES_15

```
#define TINYFORMAT_ARGTYPES_15 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15
```

Definition at line 426 of file tinyformat.h.

7.367.1.9 TINYFORMAT_ARGTYPES_16

```
#define TINYFORMAT_ARGTYPES_16 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15, class
T16
```

Definition at line 427 of file tinyformat.h.

7.367.1.10 TINYFORMAT_ARGTYPES_2

```
#define TINYFORMAT_ARGTYPES_2 class T1, class T2
```

Definition at line 413 of file tinyformat.h.

7.367.1.11 TINYFORMAT_ARGTYPES_3

```
#define TINYFORMAT_ARGTYPES_3 class T1, class T2, class T3
```

Definition at line 414 of file tinyformat.h.

7.367.1.12 TINYFORMAT_ARGTYPES_4

```
#define TINYFORMAT_ARGTYPES_4 class T1, class T2, class T3, class T4
```

Definition at line 415 of file tinyformat.h.

7.367.1.13 TINYFORMAT_ARGTYPES_5

```
#define TINYFORMAT_ARGTYPES_5 class T1, class T2, class T3, class T4, class T5
```

Definition at line 416 of file tinyformat.h.

7.367.1.14 TINYFORMAT_ARGTYPES_6

```
#define TINYFORMAT_ARGTYPES_6 class T1, class T2, class T3, class T4, class T5, class T6
```

Definition at line 417 of file tinyformat.h.

7.367.1.15 TINYFORMAT_ARGTYPES_7

```
#define TINYFORMAT_ARGTYPES_7 class T1, class T2, class T3, class T4, class T5, class T6, class T7
```

Definition at line 418 of file tinyformat.h.

7.367.1.16 TINYFORMAT_ARGTYPES_8

```
#define TINYFORMAT_ARGTYPES_8 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8
```

Definition at line 419 of file tinyformat.h.

7.367.1.17 TINYFORMAT_ARGTYPES_9

```
#define TINYFORMAT_ARGTYPES_9 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9
```

Definition at line 420 of file tinyformat.h.

7.367.1.18 TINYFORMAT_ASSERT

```
#define TINYFORMAT_ASSERT(
                cond ) assert(cond)
```

Definition at line 142 of file tinyformat.h.

7.367.1.19 TINYFORMAT_DEFINE_FORMAT_TRUNCATED_CSTR

```
#define TINYFORMAT_DEFINE_FORMAT_TRUNCATED_CSTR(
                type )
```

Value:

```
inline void formatTruncated(std::ostream& out, type* value, int ntrunc) \
{                                                                      \
    std::streamsize len = 0;                                           \
    while(len < ntrunc && value[len] != 0)                             \
        ++len;                                                         \
    out.write(value, len);                                             \
}
```

Definition at line 278 of file tinyformat.h.

7.367.1.20 TINYFORMAT_DEFINE_FORMATVALUE_CHAR

```
#define TINYFORMAT_DEFINE_FORMATVALUE_CHAR(
                charType )
```

Value:

```
inline void formatValue(std::ostream& out, const char* /*fmtBegin*/,  \
                        const char* fmtEnd, int , charType value) \
{                                                                     \
    switch(*(fmtEnd-1))                                               \
    {                                                                 \
        case 'u': case 'd': case 'i': case 'o': case 'X': case 'x':   \
            out « static_cast<int>(value); break;                    \
        default:                                                      \
            out « value;                  break;                     \
    }                                                                 \
}
```

Definition at line 347 of file tinyformat.h.

7.367.1.21 TINYFORMAT_ERROR

```
#define TINYFORMAT_ERROR(
                reason ) assert(0 && reason)
```

Definition at line 146 of file tinyformat.h.

7.367.1.22 TINYFORMAT_FOREACH_ARGNUM

```
#define TINYFORMAT_FOREACH_ARGNUM(
                m )  m(1) m(2) m(3) m(4) m(5) m(6) m(7) m(8) m(9) m(10) m(11) m(12) m(13) m(14)
m(15) m(16)
```

Definition at line 480 of file tinyformat.h.

7.367.1.23 TINYFORMAT_HIDDEN

```
#define TINYFORMAT_HIDDEN
```

Definition at line 166 of file tinyformat.h.

7.367.1.24 TINYFORMAT_MAKE_FORMAT_FUNCS

```
#define TINYFORMAT_MAKE_FORMAT_FUNCS(
                n )
```

Value:

```
                                                                          \
template<TINYFORMAT_ARGTYPES(n)>                                          \
void format(std::ostream& out, const char* fmt, TINYFORMAT_VARARGS(n))    \
{                                                                         \
    vformat(out, fmt, makeFormatList(TINYFORMAT_PASSARGS(n)));            \
}                                                                         \
                                                                          \
template<TINYFORMAT_ARGTYPES(n)>                                          \
std::string format(const char* fmt, TINYFORMAT_VARARGS(n))                \
{                                                                         \
    std::ostringstream oss;                                               \
    format(oss, fmt, TINYFORMAT_PASSARGS(n));                             \
    return oss.str();                                                     \
}                                                                         \
                                                                          \
template<TINYFORMAT_ARGTYPES(n)>                                          \
void printf(const char* fmt, TINYFORMAT_VARARGS(n))                       \
{                                                                         \
    format(std::cout, fmt, TINYFORMAT_PASSARGS(n));                       \
}                                                                         \
                                                                          \
template<TINYFORMAT_ARGTYPES(n)>                                          \
void printfln(const char* fmt, TINYFORMAT_VARARGS(n))                     \
{                                                                         \
    format(std::cout, fmt, TINYFORMAT_PASSARGS(n));                       \
    std::cout « '\n';                                                    \
}
```

Definition at line 1019 of file tinyformat.h.

7.367.1.25 TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR

```
#define TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR(
               n )
```

Value:

```
                                                                        \
        template<TINYFORMAT_ARGTYPES(n)>                                \
        FormatListN(TINYFORMAT_VARARGS(n))                              \
            : FormatList(&m_formatterStore[0], n)                       \
        { TINYFORMAT_ASSERT(n == N); init(0, TINYFORMAT_PASSARGS(n)); } \
                                                                        \
        template<TINYFORMAT_ARGTYPES(n)>                                \
        void init(int i, TINYFORMAT_VARARGS(n))                         \
        {                                                               \
            m_formatterStore[i] = FormatArg(v1);                        \
            init(i+1 TINYFORMAT_PASSARGS_TAIL(n));                      \
        }
```

Definition at line 885 of file tinyformat.h.

7.367.1.26 TINYFORMAT_MAKE_MAKEFORMATLIST

```
#define TINYFORMAT_MAKE_MAKEFORMATLIST(
               n )
```

Value:

```
template<TINYFORMAT_ARGTYPES(n)>                              \
detail::FormatListN<n> makeFormatList(TINYFORMAT_VARARGS(n))  \
{                                                             \
    return detail::FormatListN<n>(TINYFORMAT_PASSARGS(n));    \
}
```

Definition at line 939 of file tinyformat.h.

7.367.1.27 TINYFORMAT_PASSARGS

```
#define TINYFORMAT_PASSARGS(
               n ) TINYFORMAT_PASSARGS_ ## n
```

Definition at line 373 of file tinyformat.h.

7.367.1.28 TINYFORMAT_PASSARGS_1

```
#define TINYFORMAT_PASSARGS_1 v1
```

Definition at line 446 of file tinyformat.h.

7.367.1.29 TINYFORMAT_PASSARGS_10

```
#define TINYFORMAT_PASSARGS_10 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
```

Definition at line 455 of file tinyformat.h.

7.367.1.30 TINYFORMAT_PASSARGS_11

```
#define TINYFORMAT_PASSARGS_11 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
```

Definition at line 456 of file tinyformat.h.

7.367.1.31 TINYFORMAT_PASSARGS_12

```
#define TINYFORMAT_PASSARGS_12 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
```

Definition at line 457 of file tinyformat.h.

7.367.1.32 TINYFORMAT_PASSARGS_13

```
#define TINYFORMAT_PASSARGS_13 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
```

Definition at line 458 of file tinyformat.h.

7.367.1.33 TINYFORMAT_PASSARGS_14

```
#define TINYFORMAT_PASSARGS_14 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
```

Definition at line 459 of file tinyformat.h.

7.367.1.34 TINYFORMAT_PASSARGS_15

```
#define TINYFORMAT_PASSARGS_15 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
```

Definition at line 460 of file tinyformat.h.

7.367.1.35 TINYFORMAT_PASSARGS_16

```
#define TINYFORMAT_PASSARGS_16 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15,
v16
```

Definition at line 461 of file tinyformat.h.

7.367.1.36 TINYFORMAT_PASSARGS_2

```
#define TINYFORMAT_PASSARGS_2 v1, v2
```

Definition at line 447 of file tinyformat.h.

7.367.1.37 TINYFORMAT_PASSARGS_3

```
#define TINYFORMAT_PASSARGS_3 v1, v2, v3
```

Definition at line 448 of file tinyformat.h.

7.367.1.38 TINYFORMAT_PASSARGS_4

```
#define TINYFORMAT_PASSARGS_4 v1, v2, v3, v4
```

Definition at line 449 of file tinyformat.h.

7.367.1.39 TINYFORMAT_PASSARGS_5

```
#define TINYFORMAT_PASSARGS_5 v1, v2, v3, v4, v5
```

Definition at line 450 of file tinyformat.h.

7.367.1.40 TINYFORMAT_PASSARGS_6

```
#define TINYFORMAT_PASSARGS_6 v1, v2, v3, v4, v5, v6
```

Definition at line 451 of file tinyformat.h.

7.367.1.41 TINYFORMAT_PASSARGS_7

```
#define TINYFORMAT_PASSARGS_7 v1, v2, v3, v4, v5, v6, v7
```

Definition at line 452 of file tinyformat.h.

7.367.1.42 TINYFORMAT_PASSARGS_8

```
#define TINYFORMAT_PASSARGS_8 v1, v2, v3, v4, v5, v6, v7, v8
```

Definition at line 453 of file tinyformat.h.

7.367.1.43 TINYFORMAT_PASSARGS_9

```
#define TINYFORMAT_PASSARGS_9 v1, v2, v3, v4, v5, v6, v7, v8, v9
```

Definition at line 454 of file tinyformat.h.

7.367.1.44 TINYFORMAT_PASSARGS_TAIL

```
#define TINYFORMAT_PASSARGS_TAIL(
              n ) TINYFORMAT_PASSARGS_TAIL_ ## n
```

Definition at line 374 of file tinyformat.h.

7.367.1.45 TINYFORMAT_PASSARGS_TAIL_1

```
#define TINYFORMAT_PASSARGS_TAIL_1
```

Definition at line 463 of file tinyformat.h.

7.367.1.46 TINYFORMAT_PASSARGS_TAIL_10

```
#define TINYFORMAT_PASSARGS_TAIL_10 , v2, v3, v4, v5, v6, v7, v8, v9, v10
```

Definition at line 472 of file tinyformat.h.

7.367.1.47 TINYFORMAT_PASSARGS_TAIL_11

```
#define TINYFORMAT_PASSARGS_TAIL_11 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
```

Definition at line 473 of file tinyformat.h.

7.367.1.48 TINYFORMAT_PASSARGS_TAIL_12

```
#define TINYFORMAT_PASSARGS_TAIL_12 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
```

Definition at line 474 of file tinyformat.h.

7.367.1.49 TINYFORMAT_PASSARGS_TAIL_13

```
#define TINYFORMAT_PASSARGS_TAIL_13 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
```

Definition at line 475 of file tinyformat.h.

7.367.1.50 TINYFORMAT_PASSARGS_TAIL_14

```
#define TINYFORMAT_PASSARGS_TAIL_14 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
```

Definition at line 476 of file tinyformat.h.

7.367.1.51 TINYFORMAT_PASSARGS_TAIL_15

```
#define TINYFORMAT_PASSARGS_TAIL_15 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
```

Definition at line 477 of file tinyformat.h.

7.367.1.52 TINYFORMAT_PASSARGS_TAIL_16

```
#define TINYFORMAT_PASSARGS_TAIL_16 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14,
v15, v16
```

Definition at line 478 of file tinyformat.h.

7.367.1.53 TINYFORMAT_PASSARGS_TAIL_2

```
#define TINYFORMAT_PASSARGS_TAIL_2 , v2
```

Definition at line 464 of file tinyformat.h.

7.367.1.54 TINYFORMAT_PASSARGS_TAIL_3

```
#define TINYFORMAT_PASSARGS_TAIL_3 , v2, v3
```

Definition at line 465 of file tinyformat.h.

7.367.1.55 TINYFORMAT_PASSARGS_TAIL_4

```
#define TINYFORMAT_PASSARGS_TAIL_4 , v2, v3, v4
```

Definition at line 466 of file tinyformat.h.

7.367.1.56 TINYFORMAT_PASSARGS_TAIL_5

```
#define TINYFORMAT_PASSARGS_TAIL_5 , v2, v3, v4, v5
```

Definition at line 467 of file tinyformat.h.

7.367.1.57 TINYFORMAT_PASSARGS_TAIL_6

```
#define TINYFORMAT_PASSARGS_TAIL_6 , v2, v3, v4, v5, v6
```

Definition at line 468 of file tinyformat.h.

7.367.1.58 TINYFORMAT_PASSARGS_TAIL_7

```
#define TINYFORMAT_PASSARGS_TAIL_7 , v2, v3, v4, v5, v6, v7
```

Definition at line 469 of file tinyformat.h.

7.367.1.59 TINYFORMAT_PASSARGS_TAIL_8

```
#define TINYFORMAT_PASSARGS_TAIL_8 , v2, v3, v4, v5, v6, v7, v8
```

Definition at line 470 of file tinyformat.h.

7.367.1.60 TINYFORMAT_PASSARGS_TAIL_9

```
#define TINYFORMAT_PASSARGS_TAIL_9 , v2, v3, v4, v5, v6, v7, v8, v9
```

Definition at line 471 of file tinyformat.h.

7.367.1.61 TINYFORMAT_VARARGS

```
#define TINYFORMAT_VARARGS(
              n ) TINYFORMAT_VARARGS_ ## n
```

Definition at line 372 of file tinyformat.h.

7.367.1.62 TINYFORMAT_VARARGS_1

```
#define TINYFORMAT_VARARGS_1 const T1& v1
```

Definition at line 429 of file tinyformat.h.

7.367.1.63 TINYFORMAT_VARARGS_10

```
#define TINYFORMAT_VARARGS_10 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10
```

Definition at line 438 of file tinyformat.h.

7.367.1.64 TINYFORMAT_VARARGS_11

```
#define TINYFORMAT_VARARGS_11 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11
```

Definition at line 439 of file tinyformat.h.

7.367.1.65 TINYFORMAT_VARARGS_12

```
#define TINYFORMAT_VARARGS_12 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const
T12& v12
```

Definition at line 440 of file tinyformat.h.

7.367.1.66 TINYFORMAT_VARARGS_13

```
#define TINYFORMAT_VARARGS_13 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const
T12& v12, const T13& v13
```

Definition at line 441 of file tinyformat.h.

7.367.1.67 TINYFORMAT_VARARGS_14

```
#define TINYFORMAT_VARARGS_14 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const
T12& v12, const T13& v13, const T14& v14
```

Definition at line 442 of file tinyformat.h.

7.367.1.68 TINYFORMAT_VARARGS_15

```
#define TINYFORMAT_VARARGS_15 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const
T12& v12, const T13& v13, const T14& v14, const T15& v15
```

Definition at line 443 of file tinyformat.h.

7.367.1.69 TINYFORMAT_VARARGS_16

```
#define TINYFORMAT_VARARGS_16 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const
T12& v12, const T13& v13, const T14& v14, const T15& v15, const T16& v16
```

Definition at line 444 of file tinyformat.h.

7.367.1.70 TINYFORMAT_VARARGS_2

```
#define TINYFORMAT_VARARGS_2 const T1& v1, const T2& v2
```

Definition at line 430 of file tinyformat.h.

7.367.1.71 TINYFORMAT_VARARGS_3

```
#define TINYFORMAT_VARARGS_3 const T1& v1, const T2& v2, const T3& v3
```

Definition at line 431 of file tinyformat.h.

7.367.1.72 TINYFORMAT_VARARGS_4

```
#define TINYFORMAT_VARARGS_4 const T1& v1, const T2& v2, const T3& v3, const T4& v4
```

Definition at line 432 of file tinyformat.h.

7.367.1.73 TINYFORMAT_VARARGS_5

```
#define TINYFORMAT_VARARGS_5 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5
```

Definition at line 433 of file tinyformat.h.

7.367.1.74 TINYFORMAT_VARARGS_6

```
#define TINYFORMAT_VARARGS_6 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,
const T6& v6
```

Definition at line 434 of file tinyformat.h.

7.367.1.75 TINYFORMAT_VARARGS_7

```
#define TINYFORMAT_VARARGS_7 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,
const T6& v6, const T7& v7
```

Definition at line 435 of file tinyformat.h.

7.367.1.76 TINYFORMAT_VARARGS_8

```
#define TINYFORMAT_VARARGS_8 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,
const T6& v6, const T7& v7, const T8& v8
```

Definition at line 436 of file tinyformat.h.

7.367.1.77 TINYFORMAT_VARARGS_9

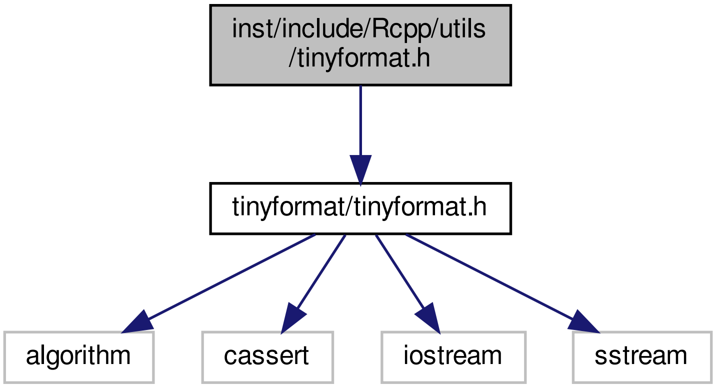
```
#define TINYFORMAT_VARARGS_9 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,
const T6& v6, const T7& v7, const T8& v8, const T9& v9
```

Definition at line 437 of file tinyformat.h.

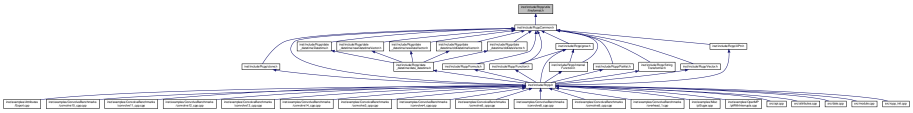
7.368 inst/include/Rcpp/utils/tinyformat.h File Reference

```
#include "tinyformat/tinyformat.h"
```

Include dependency graph for tinyformat.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

    Rcpp API.

Macros

- #define TINYFORMAT_ERROR(REASON) ::Rcpp::stop(REASON)
- #define TINYFORMAT_NO_VARIADIC_TEMPLATES
- #define TINYFORMAT_ASSERT(cond) do if (!(cond)) ::Rcpp::stop("Assertion failed"); while(0)

Functions

- void NORET Rcpp::stop (const std::string &message)

7.368.1 Macro Definition Documentation

7.368.1.1 TINYFORMAT_ASSERT

```
#define TINYFORMAT_ASSERT(
             cond ) do if (!(cond)) ::Rcpp::stop("Assertion failed"); while(0)
```

Definition at line 38 of file tinyformat.h.

7.368.1.2 TINYFORMAT_ERROR

```
#define TINYFORMAT_ERROR(
             REASON ) ::Rcpp::stop(REASON)
```

Definition at line 29 of file tinyformat.h.

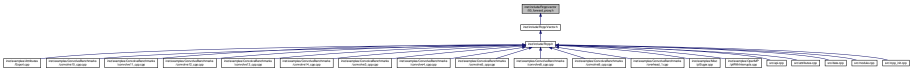
7.368.1.3 TINYFORMAT_NO_VARIADIC_TEMPLATES

```
#define TINYFORMAT_NO_VARIADIC_TEMPLATES
```

Definition at line 35 of file tinyformat.h.

7.369 inst/include/Rcpp/vector/00_forward_proxy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/00_forward_proxy.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

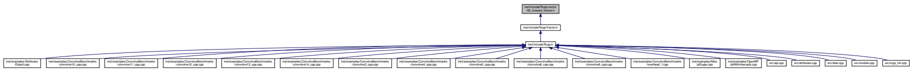
  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

7.370 inst/include/Rcpp/vector/00_forward_Vector.h File Reference

This graph shows which files directly or indirectly include this file:



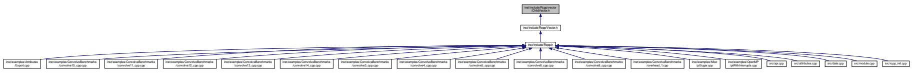
Namespaces

- Rcpp

  Rcpp API.

7.371 inst/include/Rcpp/vector/ChildVector.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::ChildVector< T >

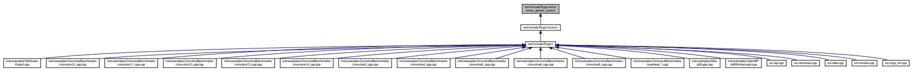
Namespaces

- Rcpp

  Rcpp API.

7.372 inst/include/Rcpp/vector/const_generic_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >

Namespaces

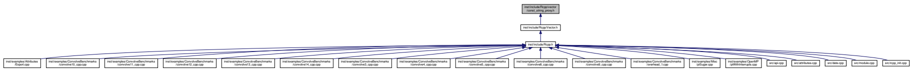
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.373 inst/include/Rcpp/vector/const_string_proxy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/const_string_proxy.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

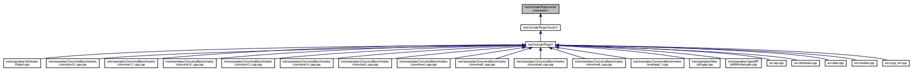
  internal implementation details

Functions

- template<int RT>
  bool Rcpp::internal::operator< (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator> (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator>= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator<= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & Rcpp::internal::operator<< (std::ostream &os, const const_string_proxy< STRSXP, StoragePolicy > &proxy)
- template<template< class > class StoragePolicy>
  std::string Rcpp::internal::operator+ (const std::string &x, const const_string_proxy< STRSXP, StoragePolicy > &y)

7.374 inst/include/Rcpp/vector/converter.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/converter.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::internal::element_converter< RTYPE >
- class Rcpp::internal::string_element_converter< RTYPE >
- class Rcpp::internal::generic_element_converter< RTYPE >
- struct Rcpp::traits::r_vector_element_converter< RTYPE >
- struct Rcpp::traits::r_vector_element_converter< STRSXP >
- struct Rcpp::traits::r_vector_element_converter< VECSXP >
- struct Rcpp::traits::r_vector_element_converter< EXPRSXP >

Namespaces

- Rcpp

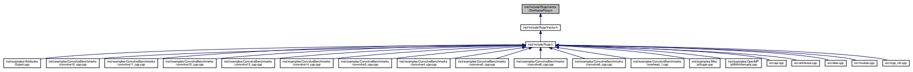
  Rcpp API.
- Rcpp::internal

  internal implementation details
- Rcpp::traits

  traits used to dispatch wrap

7.375 inst/include/Rcpp/vector/DimNameProxy.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/DimNameProxy.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::internal::DimNameProxy

Namespaces

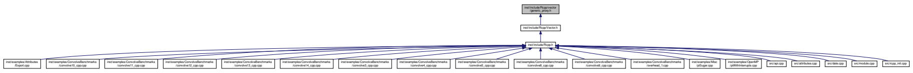
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.376 inst/include/Rcpp/vector/generic_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >

Namespaces

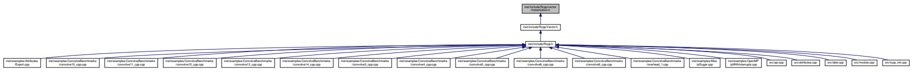
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.377 inst/include/Rcpp/vector/instantiation.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

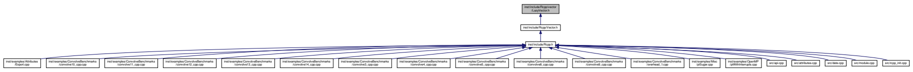
  Rcpp API.

Typedefs

- typedef Vector< CPLXSXP > Rcpp::ComplexVector
- typedef Vector< INTSXP > Rcpp::IntegerVector
- typedef Vector< LGLSXP > Rcpp::LogicalVector
- typedef Vector< REALSXP > Rcpp::NumericVector
- typedef Vector< REALSXP > Rcpp::DoubleVector
- typedef Vector< RAWSXP > Rcpp::RawVector
- typedef Vector< STRSXP > Rcpp::CharacterVector
- typedef Vector< STRSXP > Rcpp::StringVector
- typedef Vector< VECSXP > Rcpp::GenericVector
- typedef Vector< VECSXP > Rcpp::List
- typedef Vector< EXPRSXP > Rcpp::ExpressionVector
- typedef Matrix< CPLXSXP > Rcpp::ComplexMatrix
- typedef Matrix< INTSXP > Rcpp::IntegerMatrix
- typedef Matrix< LGLSXP > Rcpp::LogicalMatrix
- typedef Matrix< REALSXP > Rcpp::NumericMatrix
- typedef Matrix< RAWSXP > Rcpp::RawMatrix
- typedef Matrix< STRSXP > Rcpp::CharacterMatrix
- typedef Matrix< STRSXP > Rcpp::StringMatrix
- typedef Matrix< VECSXP > Rcpp::GenericMatrix
- typedef Matrix< VECSXP > Rcpp::ListMatrix
- typedef Matrix< EXPRSXP > Rcpp::ExpressionMatrix

7.378 inst/include/Rcpp/vector/LazyVector.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/LazyVector.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::internal::LazyVector< VECTOR >
- class Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >

Namespaces

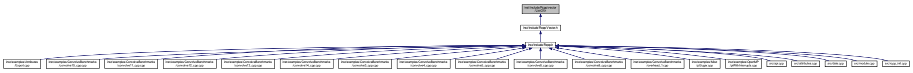
- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

7.379 inst/include/Rcpp/vector/ListOf.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::ListOf< T >

Namespaces

- Rcpp

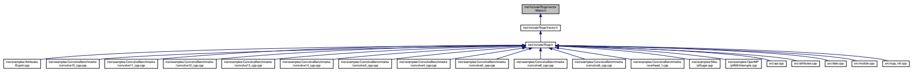
  Rcpp API.
- Rcpp::sugar

Functions

- template<typename T , typename Function >
  List Rcpp::lapply (const ListOf< T > &t, Function fun)
- template<typename T , typename Function >
  T Rcpp::sapply (const ListOf< T > &t, Function fun)

7.380 inst/include/Rcpp/vector/Matrix.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::Matrix< RTYPE, StoragePolicy >
- struct Rcpp::Matrix< RTYPE, StoragePolicy >::r_type
- struct Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na

Namespaces

- Rcpp

  Rcpp API.

Macros

- #define RCPP_GENERATE_MATRIX_SCALAR_OPERATOR(__OPERATOR__)
- #define RCPP_GENERATE_SCALAR_MATRIX_OPERATOR(__OPERATOR__)

Functions

- internal::DimNameProxy Rcpp::rownames (SEXP x)
- internal::DimNameProxy Rcpp::colnames (SEXP x)
- template<template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Matrix< REALSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Matrix< INTSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Matrix< STRSXP, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>
  std::ostream & Rcpp::operator<< (std::ostream &s, const Matrix< RTYPE, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>
  Matrix< RTYPE, StoragePolicy > Rcpp::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)

7.380.1 Macro Definition Documentation

7.380.1.1 RCPP_GENERATE_MATRIX_SCALAR_OPERATOR

```
#define RCPP_GENERATE_MATRIX_SCALAR_OPERATOR(
                __OPERATOR__ )
```

Value:

```
    template <int RTYPE, template <class> class StoragePolicy, typename T >
                    \
    inline typename traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< T
       >::type,        \
         typename Matrix<RTYPE, StoragePolicy>::stored_type >::value, Matrix<RTYPE, StoragePolicy> >::type
                    \
             operator __OPERATOR__ (const Matrix<RTYPE, StoragePolicy> &lhs, const T &rhs) {
                    \
        Vector<RTYPE, StoragePolicy> v = static_cast<const Vector<RTYPE, StoragePolicy> &>(lhs) __OPERATOR__
       rhs;            \
        v.attr("dim") = Vector<INTSXP>::create(lhs.nrow(), lhs.ncol());
                    \
        return as< Matrix<RTYPE, StoragePolicy> >(v);
                    \
    }
```

Definition at line 250 of file Matrix.h.

7.380.1.2 RCPP_GENERATE_SCALAR_MATRIX_OPERATOR

```
#define RCPP_GENERATE_SCALAR_MATRIX_OPERATOR(
                __OPERATOR__ )
```

Value:

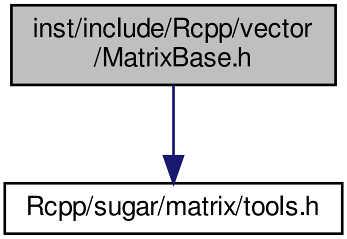
```
template <int RTYPE, template <class> class StoragePolicy, typename T >
                \
inline typename traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< T
   >::type,        \
     typename Matrix<RTYPE, StoragePolicy>::stored_type >::value, Matrix<RTYPE, StoragePolicy> >::type
                \
         operator __OPERATOR__ (const T &lhs, const Matrix<RTYPE, StoragePolicy> &rhs) {
                \
    Vector<RTYPE, StoragePolicy> v = lhs __OPERATOR__ static_cast<const Vector<RTYPE, StoragePolicy>
   &>(rhs);             \
    v.attr("dim") = Vector<INTSXP>::create(rhs.nrow(), rhs.ncol());
                \
    return as< Matrix<RTYPE, StoragePolicy> >(v);
                \
}
```

Definition at line 267 of file Matrix.h.

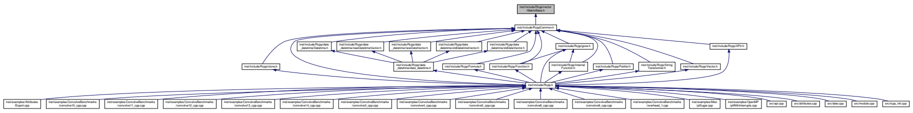
7.381 inst/include/Rcpp/vector/MatrixBase.h File Reference

```
#include <Rcpp/sugar/matrix/tools.h>
```

Include dependency graph for MatrixBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::MatrixBase< RTYPE, na, MATRIX >
- struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type
- struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface
- struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na
- class Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator

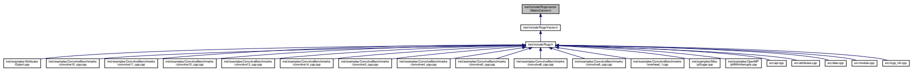
Namespaces

- Rcpp

  Rcpp API.

7.382 inst/include/Rcpp/vector/MatrixColumn.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::MatrixColumn< RTYPE >
- class Rcpp::ConstMatrixColumn< RTYPE >

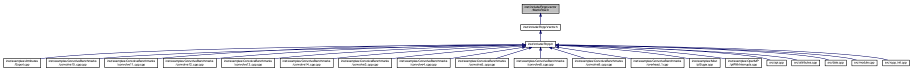
Namespaces

- Rcpp

  Rcpp API.

7.383 inst/include/Rcpp/vector/MatrixRow.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::MatrixRow< RTYPE >
- struct Rcpp::MatrixRow< RTYPE >::iter_traits
- struct Rcpp::MatrixRow< RTYPE >::const_iter_traits
- class Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >
- class Rcpp::ConstMatrixRow< RTYPE >
- class Rcpp::ConstMatrixRow< RTYPE >::const_iterator

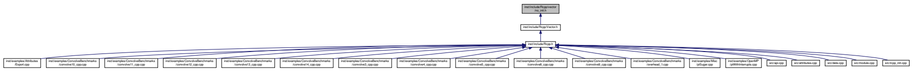
Namespaces

- Rcpp

    Rcpp API.

7.384 inst/include/Rcpp/vector/no_init.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/no_init.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::no_init_vector
- class Rcpp::no_init_matrix

Namespaces

- Rcpp

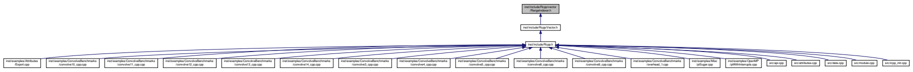
    Rcpp API.

Functions

- no_init_vector Rcpp::no_init (R_xlen_t size)
- no_init_matrix Rcpp::no_init (int nr, int nc)

7.385 inst/include/Rcpp/vector/RangeIndexer.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/RangeIndexer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Classes

- class Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Macros

- #define UNROLL_LOOP(OP)

7.385.1 Macro Definition Documentation

7.385.1.1 UNROLL_LOOP

```
#define UNROLL_LOOP(
                OP )
```

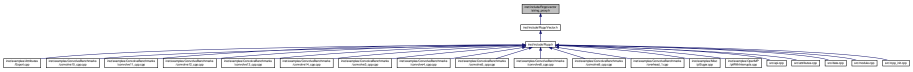
Value:

```
    typedef typename ::Rcpp::traits::Extractor<RTYPE,NA,T>::type EXT ; \
    const EXT& input( x.get_ref() ) ;                                  \
    int __trip_count = (size_) » 2;                       \
    int i=0 ;                                              \
    for ( ; __trip_count > 0 ; --__trip_count) {           \
        start[i] OP input[i] ; i++ ;                       \
        start[i] OP input[i] ; i++ ;                       \
        start[i] OP input[i] ; i++ ;                       \
        start[i] OP input[i] ; i++ ;                       \
    }                                                      \
    switch (size_ - i){                                    \
      case 3:                                              \
          start[i] OP input[i] ; i++ ;                     \
      case 2:                                              \
          start[i] OP input[i] ; i++ ;                     \
      case 1:                                              \
          start[i] OP input[i] ; i++ ;                     \
      case 0:                                              \
      default:                                             \
          return *this ;                                   \
    }
```

Definition at line 25 of file RangeIndexer.h.

7.386 inst/include/Rcpp/vector/string_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::internal::string_proxy< RTYPE, StoragePolicy >

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

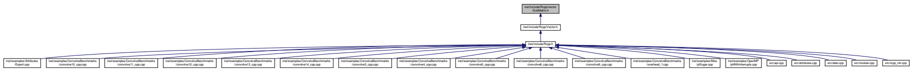
  internal implementation details

Functions

- template<int RT>
  bool Rcpp::internal::operator< (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator> (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator>= (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- template<int RT>
  bool Rcpp::internal::operator<= (const string_proxy< RT > &lhs, const string_proxy< RT > &rhs)
- std::ostream & Rcpp::internal::operator<< (std::ostream &os, const string_proxy< STRSXP > &proxy)
- std::string Rcpp::internal::operator+ (const std::string &x, const string_proxy< STRSXP > &y)

7.387 inst/include/Rcpp/vector/SubMatrix.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::SubMatrix< RTYPE >

Namespaces

- Rcpp

  Rcpp API.

Macros

- #define RCPP_WRAP_SUBMATRIX(RTYPE)

7.387.1 Macro Definition Documentation

7.387.1.1 RCPP_WRAP_SUBMATRIX

```
#define RCPP_WRAP_SUBMATRIX(
                RTYPE )
```

Value:

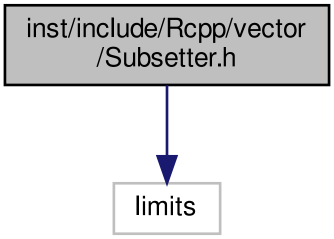
```
template<> inline SEXP wrap< SubMatrix<RTYPE> >(  \
    const SubMatrix<RTYPE>& object               \
    ) {                                          \
        return Matrix<RTYPE>( object ) ;         \
    }
```

Definition at line 90 of file SubMatrix.h.

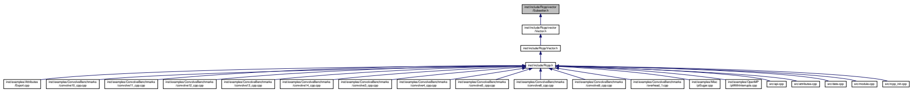
7.388 inst/include/Rcpp/vector/Subsetter.h File Reference

`#include <limits>`

Include dependency graph for Subsetter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >

Namespaces

- Rcpp

    Rcpp API.

Macros

- #define RCPP_GENERATE_SUBSET_PROXY_OPERATOR(__OPERATOR__)

7.388.1 Macro Definition Documentation

7.388.1.1 RCPP_GENERATE_SUBSET_PROXY_OPERATOR

```
#define RCPP_GENERATE_SUBSET_PROXY_OPERATOR(
               __OPERATOR__ )
```

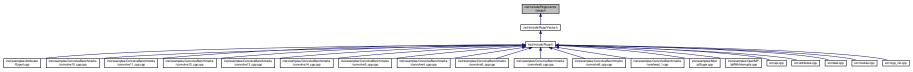
Value:

```
    template <int RTYPE_OTHER, template <class> class StoragePolicyOther,             \
              int RHS_RTYPE_OTHER, bool RHS_NA_OTHER, typename RHS_T_OTHER>           \
    Vector<RTYPE, StoragePolicy> operator __OPERATOR__ (                              \
        const SubsetProxy<RTYPE_OTHER, StoragePolicyOther, RHS_RTYPE_OTHER,           \
                          RHS_NA_OTHER, RHS_T_OTHER>& other) {                        \
        Vector<RTYPE, StoragePolicy> result(indices_n);                               \
        if (other.indices_n == 1) {                                                   \
            for (R_xlen_t i = 0; i < indices_n; ++i)                                  \
                result[i] = lhs[indices[i]] __OPERATOR__ other.lhs[other.indices[0]]; \
        } else if (indices_n == other.indices_n) {                                    \
            for (R_xlen_t i = 0; i < indices_n; ++i)                                  \
                result[i] = lhs[indices[i]] __OPERATOR__ other.lhs[other.indices[i]]; \
        } else {                                                                      \
            stop("index error");                                                      \
        }                                                                             \
        return result;                                                                \
    }
```

Definition at line 237 of file Subsetter.h.

7.389 inst/include/Rcpp/vector/swap.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/swap.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- std

Macros

- #define RCPP_GENERATE_SWAP(TYPE, RTYPE)

7.389.1 Macro Definition Documentation

7.389.1.1 RCPP_GENERATE_SWAP

```
#define RCPP_GENERATE_SWAP(
                TYPE,
                RTYPE )
```

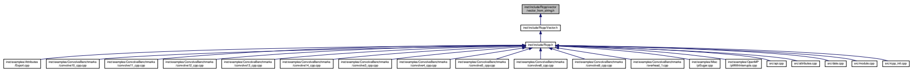
Value:

```
        template<> inline void swap< Rcpp::internal::TYPE<RTYPE> >(      \
                Rcpp::internal::TYPE<RTYPE>& a ,                                                \
                Rcpp::internal::TYPE<RTYPE>& b) {                                               \
                        a.swap(b) ;
                    \
                }
```

Definition at line 28 of file swap.h.

7.390 inst/include/Rcpp/vector/vector_from_string.h File Reference

This graph shows which files directly or indirectly include this file:


inst/include/Rcpp/vector/vector_from_string.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

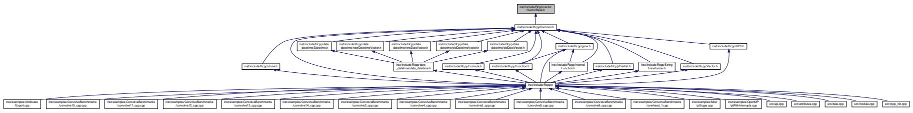
  internal implementation details

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)

7.391 inst/include/Rcpp/vector/VectorBase.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::VectorBase< RTYPE, na, VECTOR >
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits
- struct Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits
- class Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >

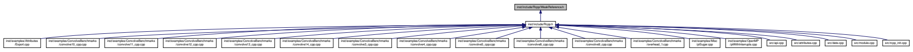
Namespaces

- Rcpp

  Rcpp API.

7.392 inst/include/Rcpp/WeakReference.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- Rcpp

  Rcpp API.

Typedefs

- typedef WeakReference_Impl< PreserveStorage > Rcpp::WeakReference

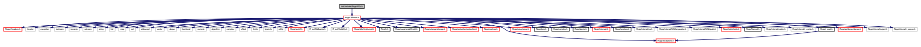
Functions

- Rcpp::RCPP_API_CLASS (WeakReference_Impl)

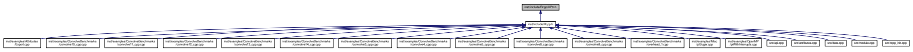
7.393 inst/include/Rcpp/XPtr.h File Reference

`#include <RcppCommon.h>`

Include dependency graph for XPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >

Namespaces

- Rcpp

  Rcpp API.

Functions

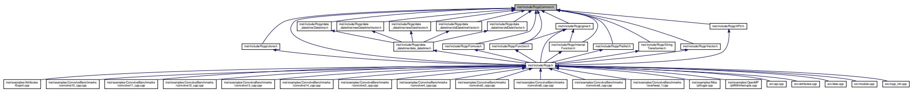
- template<typename T >
  void Rcpp::standard_delete_finalizer (T ∗obj)
- template<typename T , void Finalizer>
  void Rcpp::finalizer_wrapper (SEXP p)

7.394 inst/include/RcppCommon.h File Reference

```
#include <Rcpp/r/headers.h>
#include <iterator>
#include <exception>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <list>
#include <map>
#include <set>
#include <stdexcept>
#include <vector>
#include <deque>
#include <functional>
#include <numeric>
#include <algorithm>
#include <complex>
#include <cfloat>
#include <limits>
#include <typeinfo>
#include <utility>
#include <Rcpp/sprintf.h>
#include <R_ext/Callbacks.h>
#include <R_ext/Visibility.h>
#include <Rcpp/utils/tinyformat.h>
#include <Rmath.h>
#include <Rcpp/sugar/undoRmath.h>
#include <Rcpp/storage/storage.h>
#include <Rcpp/protection/protection.h>
#include <Rcpp/routines.h>
#include <Rcpp/exceptions.h>
#include <Rcpp/proxy/proxy.h>
#include <Rcpp/lang.h>
#include <Rcpp/complex.h>
#include <Rcpp/barrier.h>
#include <Rcpp/Interrupt.h>
#include <Rcpp/longlong.h>
#include <Rcpp/internal/na.h>
#include <Rcpp/internal/NAComparator.h>
#include <Rcpp/internal/NAEquals.h>
#include <Rcpp/traits/traits.h>
#include <Rcpp/Named.h>
#include <Rcpp/internal/caster.h>
#include <Rcpp/internal/r_vector.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/api/bones/bones.h>
#include <Rcpp/internal/export.h>
#include <Rcpp/internal/r_coerce.h>
#include <Rcpp/as.h>
#include <Rcpp/InputParameter.h>
#include <Rcpp/is.h>
#include <Rcpp/vector/VectorBase.h>
```

```
#include <Rcpp/vector/MatrixBase.h>
#include <Rcpp/internal/ListInitialization.h>
#include <Rcpp/internal/Proxy_Iterator.h>
#include <Rcpp/internal/SEXP_Iterator.h>
#include <Rcpp/internal/converter.h>
#include <Rcpp/print.h>
#include <Rcpp/algo.h>
#include <Rcpp/sugar/sugar_forward.h>
#include <Rcpp/iostream/Rstreambuf.h>
#include <Rcpp/internal/wrap.h>
```

This graph shows which files directly or indirectly include this file:


inst/include/RcppCommon.h
inst/include/Rcpp/date_datetime/Datetime.h
inst/include/Rcpp/date_datetime/newDatetimeVector.h
inst/include/Rcpp/date_datetime/newDateVector.h
inst/include/Rcpp/date_datetime/oldDatetimeVector.h
inst/include/Rcpp/date_datetime/oldDateVector.h
inst/include/Rcpp/grow.h
inst/include/Rcpp/XPtr.h
inst/include/Rcpp/clone.h
inst/include/Rcpp/date_datetime/date_datetime.h
inst/include/Rcpp/Formula.h
inst/include/Rcpp/Function.h
inst/include/Rcpp/Internal Function.h
inst/include/Rcpp/Pairlist.h
inst/include/Rcpp/String Transformer.h
inst/include/Rcpp/Vector.h
inst/include/Rcpp.h
inst/examples/Attributes/Export.cpp
inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp
inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp
inst/examples/ConvolveBenchmarks/overhead_1.cpp
inst/examples/Misc/piSugar.cpp
inst/examples/OpenMP/piWithInterrupts.cpp
src/api.cpp
src/attributes.cpp
src/date.cpp
src/module.cpp
src/rcpp_init.cpp

Namespaces

- Rcpp

  Rcpp API.
- Rcpp::traits

  traits used to dispatch wrap
- Rcpp::internal

  internal implementation details

Macros

- #define RcppExport extern "C" attribute_visible

Functions

- SEXP Rcpp::Rcpp_fast_eval (SEXP expr, SEXP env)
- SEXP Rcpp::Rcpp_eval (SEXP expr, SEXP env)
- attribute_hidden SEXP Rcpp::Rcpp_precious_preserve (SEXP object)
- attribute_hidden void Rcpp::Rcpp_precious_remove (SEXP token)
- SEXP Rcpp::internal::Rcpp_eval_impl (SEXP expr, SEXP env)
- SEXP Rcpp::Rcpp_PreserveObject (SEXP x)
- void Rcpp::Rcpp_ReleaseObject (SEXP x)
- SEXP Rcpp::Rcpp_ReplaceObject (SEXP x, SEXP y)
- SEXP Rcpp::Rcpp_PreciousPreserve (SEXP object)
- void Rcpp::Rcpp_PreciousRelease (SEXP token)
- template<typename Class >
  SEXP Rcpp::internal::make_new_object (Class ∗ptr)

7.394.1 Macro Definition Documentation

7.394.1.1 RcppExport

```
#define RcppExport extern "C" attribute_visible
```

Examples

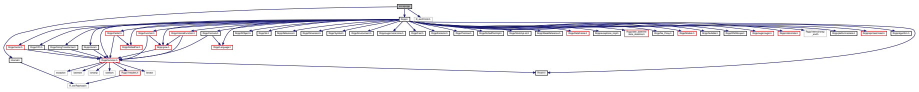
ConvolveBenchmarks/convolve3_cpp.cpp, and ConvolveBenchmarks/convolve4_cpp.cpp.

Definition at line 140 of file RcppCommon.h.

7.395 src/api.cpp File Reference

```
#include <Rcpp.h>
#include "internal.h"
#include <R_ext/PrtUtil.h>
```

Include dependency graph for api.cpp:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Macros

- #define COMPILING_RCPP
- #define RCPP_USE_GLOBAL_ROSTREAM

Functions

- attribute_hidden Rostream< true > & Rcpp::Rcpp_cout_get ()
- attribute_hidden Rostream< false > & Rcpp::Rcpp_cerr_get ()
- attribute_hidden unsigned long Rcpp::internal::enterRNGScope ()
- attribute_hidden unsigned long Rcpp::internal::exitRNGScope ()
- attribute_hidden unsigned long Rcpp::internal::beginSuspendRNGSynchronization ()
- attribute_hidden unsigned long Rcpp::internal::endSuspendRNGSynchronization ()
- attribute_hidden char ∗ Rcpp::internal::get_string_buffer ()
- attribute_hidden const char ∗ Rcpp::type2name (SEXP x)
- std::string demangle (const std::string &name)
- const char ∗ short_file_name (const char ∗file)
- SEXP as_character_externalptr (SEXP xp)
- SEXP rcpp_capabilities ()
- SEXP rcpp_can_use_cxx0x ()
- SEXP rcpp_can_use_cxx11 ()
- SEXP stack_trace (const char ∗file, int line)
- SEXP getRcppVersionStrings ()

Variables

- int Rcpp::internal::rngSynchronizationSuspended = 0

7.395.1 Macro Definition Documentation

7.395.1.1 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 23 of file api.cpp.

7.395.1.2 RCPP_USE_GLOBAL_ROSTREAM

```
#define RCPP_USE_GLOBAL_ROSTREAM
```

Definition at line 24 of file api.cpp.

7.395.2 Function Documentation

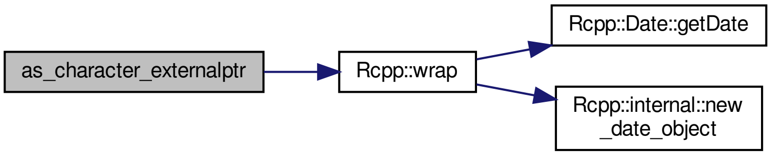
7.395.2.1 as_character_externalptr()

```
SEXP as_character_externalptr (
             SEXP xp )
```

Definition at line 160 of file api.cpp.

References Rcpp::wrap().

Here is the call graph for this function:



7.395.2.2 demangle()

```
std::string demangle (
            const std::string & name )  [inline]
```

Definition at line 125 of file api.cpp.

References GET_CALLABLE.

Referenced by exception_to_condition_template(), Rcpp::get_return_type_dispatch(), and registerFunctions().

7.395.2.3 getRcppVersionStrings()

```
SEXP getRcppVersionStrings ( )
```

Definition at line 289 of file api.cpp.

References RCPP_DEV_VERSION_STRING, and RCPP_VERSION_STRING.

7.395.2.4 rcpp_can_use_cxx0x()

```
SEXP rcpp_can_use_cxx0x ( )
```

Definition at line 255 of file api.cpp.

7.395.2.5 rcpp_can_use_cxx11()

```
SEXP rcpp_can_use_cxx11 ( )
```

Definition at line 265 of file api.cpp.

7.395.2.6 rcpp_capabilities()

```
SEXP rcpp_capabilities ( )
```

Definition at line 167 of file api.cpp.

7.395.2.7 short_file_name()

```
const char* short_file_name (
             const char * file )
```

Definition at line 149 of file api.cpp.

7.395.2.8 stack_trace()

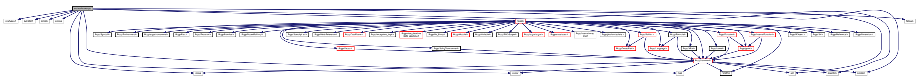
```
SEXP stack_trace (
             const char * file,
             int line )
```

Definition at line 275 of file api.cpp.

7.396 src/attributes.cpp File Reference

```
#include <sys/types.h>
#include <sys/stat.h>
#include <errno.h>
#include <cstring>
#include <string>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
#include <fstream>
#include <sstream>
#include <Rcpp.h>
```

Include dependency graph for attributes.cpp:



Classes

- class Rcpp::attributes::FileInfo
- class Rcpp::attributes::Type
- class Rcpp::attributes::Argument
- class Rcpp::attributes::Function
- class Rcpp::attributes::Param
- class Rcpp::attributes::Attribute
- class Rcpp::attributes::SourceFileAttributes
- class Rcpp::attributes::CommentState

- class Rcpp::attributes::SourceFileAttributesParser
- class Rcpp::attributes::ExportsGenerator
- class Rcpp::attributes::CppExportsGenerator
- class Rcpp::attributes::CppExportsIncludeGenerator
- class Rcpp::attributes::CppPackageIncludeGenerator
- class Rcpp::attributes::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)
- bool Rcpp::attributes::endsWith (const std::string &str, const std::string &suffix)
- 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)
- bool Rcpp::attributes::checkRSignature (const Function &function, std::string args)
- void Rcpp::attributes::initializeGlobals (std::ostream &ostr)
- void Rcpp::attributes::generateCpp (std::ostream &ostr, const SourceFileAttributes &attributes, bool include↩ Prototype, bool cppInterface, const std::string &contextId)
- void Rcpp::attributes::printArgument (std::ostream &os, const Argument &argument, bool printDefault=true)
- void Rcpp::attributes::printFunction (std::ostream &os, const Function &function, bool printArgDefaults=true)
- RcppExport SEXP sourceCppContext (SEXP sFile, SEXP sCode, SEXP sRebuild, SEXP sCacheDir, SEXP s↩ Platform)
- RcppExport SEXP compileAttributes (SEXP sPackageDir, SEXP sPackageName, SEXP sDepends, SEXP s↩ Registration, SEXP sCppFiles, SEXP sCppFileBasenames, SEXP sIncludes, SEXP sVerbose, SEXP sPlatform)

Variables

- const char ∗const Rcpp::attributes::kWhitespaceChars = " \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::kExportInvisible = "invisible"
- const char ∗const Rcpp::attributes::kExportSignature = "signature"
- const char ∗const Rcpp::attributes::kInitAttribute = "init"
- const char ∗const Rcpp::attributes::kDependsAttribute = "depends"
- const char ∗const Rcpp::attributes::kPluginsAttribute = "plugins"
- const char ∗const Rcpp::attributes::kInterfacesAttribute = "interfaces"
- const char ∗const Rcpp::attributes::kInterfaceR = "r"
- const char ∗const Rcpp::attributes::kInterfaceCpp = "cpp"
- const char ∗const Rcpp::attributes::kParamValueFalse = "false"
- const char ∗const Rcpp::attributes::kParamValueTrue = "true"
- const char ∗const Rcpp::attributes::kParamValueFALSE = "FALSE"
- const char ∗const Rcpp::attributes::kParamValueTRUE = "TRUE"
- const char ∗const Rcpp::attributes::kParamBlockStart = "{;"
- const char ∗const Rcpp::attributes::kParamBlockEnd = "}"

7.396.1 Macro Definition Documentation

7.396.1.1 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file attributes.cpp.

7.396.1.2 RCPP_NO_SUGAR

```
#define RCPP_NO_SUGAR
```

Definition at line 38 of file attributes.cpp.

7.396.2 Function Documentation

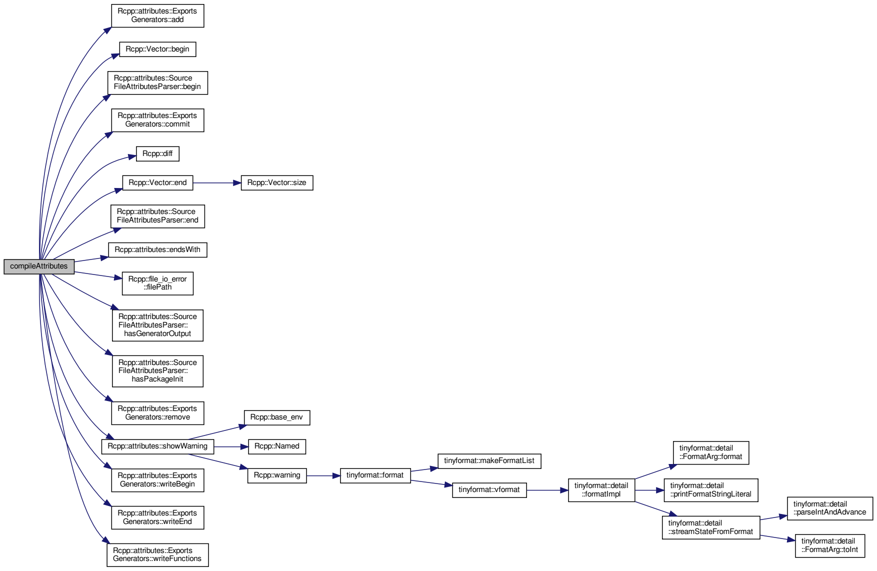
7.396.2.1 compileAttributes()

```
RcppExport SEXP compileAttributes (
              SEXP sPackageDir,
              SEXP sPackageName,
              SEXP sDepends,
              SEXP sRegistration,
              SEXP sCppFiles,
              SEXP sCppFileBasenames,
              SEXP sIncludes,
              SEXP sVerbose,
              SEXP sPlatform )
```

Definition at line 3620 of file attributes.cpp.

References Rcpp::attributes::ExportsGenerators::add(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::attributes::SourceFileAttributesParser::begin(), BEGIN_RCPP, Rcpp::attributes::ExportsGenerators::commit(), Rcpp::diff(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::attributes::SourceFileAttributesParser::end(), END_RCPP, Rcpp::attributes::endsWith(), Rcpp::file_io_error::filePath(), Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput(), Rcpp::attributes::SourceFileAttributesParser::hasPackageInit(), Rcpp::attributes::kDependsAttribute, Rcpp::Rcout, Rcpp::attributes::ExportsGenerators::remove(), Rcpp::attributes::showWarning(), Rcpp::attributes::ExportsGenerators::writeBegin(), Rcpp::attributes::ExportsGenerators::writeEnd(), and Rcpp::attributes::ExportsGenerators::writeFunctions().

Here is the call graph for this function:


Rcpp::attributes::Exports Generators::add
Rcpp::Vector::begin
Rcpp::attributes::Source FileAttributesParser::begin
Rcpp::attributes::Exports Generators::commit
Rcpp::diff
Rcpp::Vector::end
Rcpp::Vector::size
Rcpp::attributes::Source FileAttributesParser::end
Rcpp::attributes::endsWith
compileAttributes
Rcpp::file_io_error ::filePath
Rcpp::attributes::Source FileAttributesParser:: hasGeneratorOutput
Rcpp::attributes::Source FileAttributesParser:: hasPackageInit
Rcpp::attributes::Exports Generators::remove
Rcpp::base_env
Rcpp::attributes::showWarning
Rcpp::Named
Rcpp::warning
tinyformat::format
tinyformat::makeFormatList
tinyformat::vformat
tinyformat::detail ::formatImpl
tinyformat::detail ::FormatArg::format
tinyformat::detail ::printFormatStringLiteral
tinyformat::detail ::streamStateFromFormat
tinyformat::detail ::parseIntAndAdvance
tinyformat::detail ::FormatArg::toInt
Rcpp::attributes::Exports Generators::writeBegin
Rcpp::attributes::Exports Generators::writeEnd
Rcpp::attributes::Exports Generators::writeFunctions

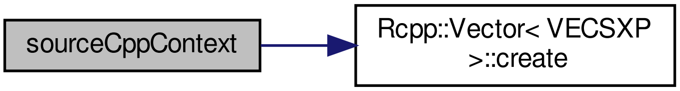
7.396.2.2 sourceCppContext()

```
RcppExport SEXP sourceCppContext (
             SEXP sFile,
             SEXP sCode,
             SEXP sRebuild,
             SEXP sCacheDir,
             SEXP sPlatform )
```

Definition at line 3556 of file attributes.cpp.

References Rcpp::_, BEGIN_RCPP, Rcpp::Vector< VECSXP >::create(), and END_RCPP.

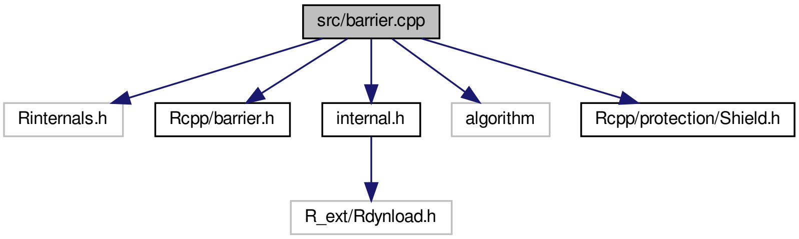
Here is the call graph for this function:



7.397 src/barrier.cpp File Reference

```
#include <Rinternals.h>
#include <Rcpp/barrier.h>
#include "internal.h"
#include <algorithm>
#include <Rcpp/protection/Shield.h>
```

Include dependency graph for barrier.cpp:



Namespaces

- Rcpp

  Rcpp API.
- Rcpp::internal

  internal implementation details

Macros

- #define COMPILING_RCPP
- #define USE_RINTERNALS
- #define RCPP_HASH_CACHE_INDEX 4
- #define RCPP_CACHE_SIZE 5
- #define RCPP_HASH_CACHE_INITIAL_SIZE 1024

Functions

- SEXP get_string_elt (SEXP x, R_xlen_t i)
- const char ∗ char_get_string_elt (SEXP x, R_xlen_t i)
- void set_string_elt (SEXP x, R_xlen_t i, SEXP value)
- void char_set_string_elt (SEXP x, R_xlen_t i, const char ∗value)
- SEXP ∗ get_string_ptr (SEXP x)
- SEXP get_vector_elt (SEXP x, R_xlen_t i)
- void set_vector_elt (SEXP x, R_xlen_t i, SEXP value)
- SEXP ∗ get_vector_ptr (SEXP x)
- void ∗ dataptr (SEXP x)
- const char ∗ char_nocheck (SEXP x)
- attribute_hidden void Rcpp::Rcpp_precious_init ()
- attribute_hidden void Rcpp::Rcpp_precious_teardown ()
- attribute_hidden SEXP Rcpp::Rcpp_precious_preserve (SEXP object)
- attribute_hidden void Rcpp::Rcpp_precious_remove (SEXP token)
- SEXP get_rcpp_cache ()
- attribute_hidden SEXP Rcpp::internal::get_Rcpp_namespace ()
- SEXP rcpp_get_stack_trace ()
- SEXP rcpp_set_stack_trace (SEXP e)
- SEXP set_error_occured (SEXP cache, SEXP e)
- SEXP set_current_error (SEXP cache, SEXP e)
- SEXP init_Rcpp_cache ()
- SEXP reset_current_error ()
- int error_occured ()
- SEXP rcpp_error_recorder (SEXP e)
- SEXP rcpp_get_current_error ()
- int ∗ get_cache (int m)

Variables

- static bool Rcpp_cache_know = false
- static SEXP Rcpp_cache = R_NilValue
- static SEXP Rcpp::Rcpp_precious = R_NilValue

7.397.1 Macro Definition Documentation

7.397.1.1 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file barrier.cpp.

7.397.1.2 RCPP_CACHE_SIZE

```
#define RCPP_CACHE_SIZE 5
```

Definition at line 85 of file barrier.cpp.

7.397.1.3 RCPP_HASH_CACHE_INDEX

```
#define RCPP_HASH_CACHE_INDEX 4
```

Definition at line 84 of file barrier.cpp.

7.397.1.4 RCPP_HASH_CACHE_INITIAL_SIZE

```
#define RCPP_HASH_CACHE_INITIAL_SIZE 1024
```

Definition at line 88 of file barrier.cpp.

7.397.1.5 USE_RINTERNALS

```
#define USE_RINTERNALS
```

Definition at line 24 of file barrier.cpp.

7.397.2 Function Documentation

7.397.2.1 char_get_string_elt()

```
const char* char_get_string_elt (
            SEXP x,
            R_xlen_t i )  [inline]
```

Definition at line 37 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.2 char_nocheck()

```
const char* char_nocheck (
            SEXP x )  [inline]
```

Definition at line 77 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions(), and Rcpp::String::setBuffer().

7.397.2.3 char_set_string_elt()

```
void char_set_string_elt (
            SEXP x,
            R_xlen_t i,
            const char * value )  [inline]
```

Definition at line 47 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.4 dataptr()

```
void* dataptr (
             SEXP x )  [inline]
```

Definition at line 72 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

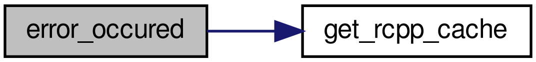
7.397.2.5 error_occured()

```
int error_occured ( )
```

Definition at line 213 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:



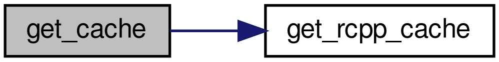
7.397.2.6 get_cache()

```
int* get_cache (
             int m )
```

Definition at line 237 of file barrier.cpp.

References get_rcpp_cache(), and RCPP_HASH_CACHE_INDEX.

Here is the call graph for this function:



7.397.2.7 get_rcpp_cache()

```
SEXP get_rcpp_cache ( )
```

Definition at line 133 of file barrier.cpp.

References Rcpp_cache, and Rcpp_cache_know.

Referenced by error_occured(), get_cache(), rcpp_error_recorder(), rcpp_get_current_error(), rcpp_get_stack_trace(), rcpp_set_stack_trace(), and reset_current_error().

7.397.2.8 get_string_elt()

```
SEXP get_string_elt (
             SEXP x,
             R_xlen_t i )  [inline]
```

Definition at line 32 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.9 get_string_ptr()

```
SEXP* get_string_ptr (
             SEXP x )  [inline]
```

Definition at line 52 of file barrier.cpp.

References GET_CALLABLE.

Referenced by Rcpp::sugar::get_const_begin(), and registerFunctions().

7.397.2.10 get_vector_elt()

```
SEXP get_vector_elt (
             SEXP x,
             R_xlen_t i )  [inline]
```

Definition at line 57 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.11 get_vector_ptr()

```
SEXP* get_vector_ptr (
            SEXP x )  [inline]
```

Definition at line 67 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.12 init_Rcpp_cache()

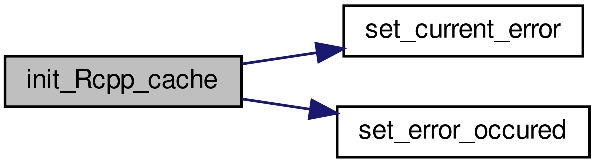
```
SEXP init_Rcpp_cache ( )
```

Definition at line 177 of file barrier.cpp.

References RCPP_CACHE_SIZE, RCPP_HASH_CACHE_INDEX, RCPP_HASH_CACHE_INITIAL_SIZE, set_↩ current_error(), and set_error_occured().

Referenced by R_init_Rcpp().

Here is the call graph for this function:



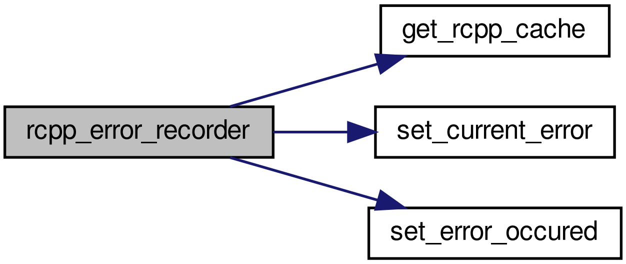
7.397.2.13 rcpp_error_recorder()

```
SEXP rcpp_error_recorder (
            SEXP e )
```

Definition at line 219 of file barrier.cpp.

References get_rcpp_cache(), set_current_error(), and set_error_occured().

Here is the call graph for this function:



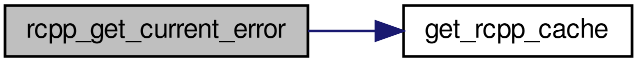
7.397.2.14 rcpp_get_current_error()

```
SEXP rcpp_get_current_error ( )
```

Definition at line 232 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:



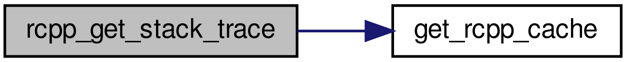
7.397.2.15 rcpp_get_stack_trace()

```
SEXP rcpp_get_stack_trace ( )
```

Definition at line 157 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:



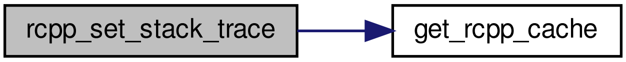
7.397.2.16 rcpp_set_stack_trace()

```
SEXP rcpp_set_stack_trace (
             SEXP e )
```

Definition at line 162 of file barrier.cpp.

References get_rcpp_cache().

Here is the call graph for this function:



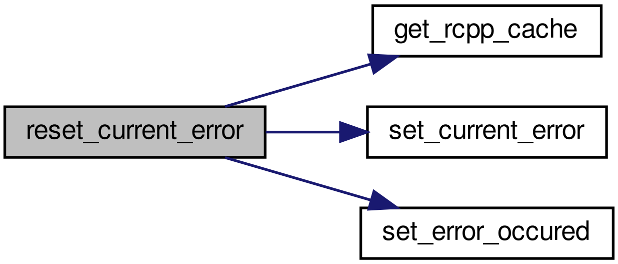
7.397.2.17 reset_current_error()

```
SEXP reset_current_error ( )
```

Definition at line 197 of file barrier.cpp.

References get_rcpp_cache(), set_current_error(), and set_error_occured().

Here is the call graph for this function:



7.397.2.18 set_current_error()

```
SEXP set_current_error (
            SEXP cache,
            SEXP e )
```

Definition at line 172 of file barrier.cpp.

Referenced by init_Rcpp_cache(), rcpp_error_recorder(), and reset_current_error().

7.397.2.19 set_error_occured()

```
SEXP set_error_occured (
            SEXP cache,
            SEXP e )
```

Definition at line 167 of file barrier.cpp.

Referenced by init_Rcpp_cache(), rcpp_error_recorder(), and reset_current_error().

7.397.2.20 set_string_elt()

```
void set_string_elt (
            SEXP x,
            R_xlen_t i,
            SEXP value )  [inline]
```

Definition at line 42 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.2.21 set_vector_elt()

```
void set_vector_elt (
            SEXP x,
            R_xlen_t i,
            SEXP value )  [inline]
```

Definition at line 62 of file barrier.cpp.

References GET_CALLABLE.

Referenced by registerFunctions().

7.397.3 Variable Documentation

7.397.3.1 Rcpp_cache

```
SEXP Rcpp_cache = R_NilValue  [static]
```

Definition at line 82 of file barrier.cpp.

Referenced by get_rcpp_cache().

7.397.3.2 Rcpp_cache_know

```
bool Rcpp_cache_know = false  [static]
```

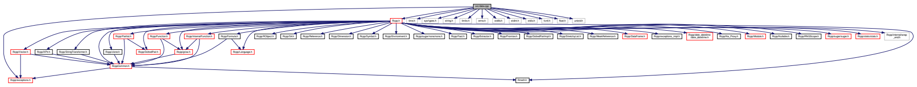
Definition at line 81 of file barrier.cpp.

Referenced by get_rcpp_cache().

7.398 src/date.cpp File Reference

```
#include <Rcpp.h>
#include <time.h>
#include <Rcpp/exceptions.h>
#include "sys/types.h"
#include "string.h"
#include "limits.h"
#include <errno.h>
#include "stdlib.h"
#include "stdint.h"
#include "stdio.h"
#include "fcntl.h"
#include "float.h"
#include <unistd.h>
```

Include dependency graph for date.cpp:



Classes

- struct Rcpp::tzhead
- struct Rcpp::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 TWOS_COMPLEMENT(t) ((t) ∼ (t) 0 < 0)
- #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 MAXVAL(t, b)
- #define MINVAL(t, b) ((t) (TYPE_SIGNED(t) ? - TWOS_COMPLEMENT(t) - MAXVAL(t, b) : 0))
- #define TZFILE_H
- #define TZDIR "/usr/local/etc/zoneinfo" /∗ Time zone object file directory ∗/
- #define TZDEFAULT "localtime"
- #define TZDEFRULES "America/New_York"
- #define TZ_MAGIC "TZif"
- #define TZ_MAX_TIMES 1200
- #define TZ_MAX_TYPES 256 /∗ Limited by what (unsigned char)'s can hold ∗/
- #define TZ_MAX_CHARS 100 /∗ Maximum number of abbreviation characters ∗/
- #define TZ_MAX_LEAPS 50 /∗ Maximum number of leap second corrections ∗/
- #define SECSPERMIN 60
- #define MINSPERHOUR 60
- #define HOURSPERDAY 24
- #define DAYSPERWEEK 7
- #define DAYSPERNYEAR 365
- #define DAYSPERLYEAR 366
- #define SECSPERHOUR (SECSPERMIN ∗ MINSPERHOUR)
- #define SECSPERDAY ((int_fast32_t) SECSPERHOUR ∗ HOURSPERDAY)
- #define MONSPERYEAR 12
- #define TM_SUNDAY 0
- #define TM_MONDAY 1
- #define TM_TUESDAY 2
- #define TM_WEDNESDAY 3
- #define TM_THURSDAY 4
- #define TM_FRIDAY 5
- #define TM_SATURDAY 6
- #define TM_JANUARY 0
- #define TM_FEBRUARY 1
- #define TM_MARCH 2
- #define TM_APRIL 3
- #define TM_MAY 4
- #define TM_JUNE 5
- #define TM_JULY 6
- #define TM_AUGUST 7
- #define TM_SEPTEMBER 8
- #define TM_OCTOBER 9
- #define TM_NOVEMBER 10
- #define TM_DECEMBER 11
- #define TM_YEAR_BASE 1900
- #define EPOCH_YEAR 1970
- #define EPOCH_WDAY TM_THURSDAY
- #define isleap(y) (((y) % 4) == 0 && (((y) % 100) != 0 || ((y) % 400) == 0))
- #define isleap_sum(a, b) isleap((a) % 400 + (b) % 400)
- #define OPEN_MODE O_RDONLY
- #define TZDEFRULESTRING ",M4.1.0,M10.5.0"

- #define BIGGEST(a, b) (((a) > (b)) ? (a) : (b))
- #define MY_TZNAME_MAX 255
- #define JULIAN_DAY 0 /∗ Jn - Julian day ∗/
- #define DAY_OF_YEAR 1 /∗ n - day of year ∗/
- #define MONTH_NTH_DAY_OF_WEEK 2 /∗ Mm.n.d - month, week, day of week ∗/
- #define gmtptr (&gmtmem)

Functions

- attribute_hidden double Rcpp::mktime00 (struct tm &tm)
- static int Rcpp::tzparse (const char ∗name, struct state ∗sp, int lastditch)
- static int Rcpp::typesequiv (const struct state ∗sp, int a, int b)
- static const char ∗ Rcpp::getsecs (const char ∗strp, int_fast32_t ∗secsp)
- static const char ∗ Rcpp::getnum (const char ∗strp, int ∗const nump, const int min, const int max)
- static const char ∗ Rcpp::getrule (const char ∗strp, struct rule ∗const rulep)
- static int_fast32_t Rcpp::transtime (int year, const struct rule ∗rulep, int_fast32_t offset)
- static struct tm ∗ Rcpp::timesub (const time_t ∗timep, int_fast32_t offset, const struct state ∗sp, struct tm ∗tmp)
- static int Rcpp::leaps_thru_end_of (const int y)
- static int Rcpp::increment_overflow (int ∗const ip, int j)
- static int Rcpp::increment_overflow_time (time_t ∗tp, int_fast32_t j)
- static int_fast32_t Rcpp::detzcode (const char ∗const codep)
- static int_fast64_t Rcpp::detzcode64 (const char ∗const codep)
- static int Rcpp::differ_by_repeat (const time_t t1, const time_t t0)
- static const char ∗ Rcpp::getzname (const char ∗strp)
- static const char ∗ Rcpp::getqzname (const char ∗strp, const int delim)
- static const char ∗ Rcpp::getoffset (const char ∗strp, int_fast32_t ∗const offsetp)
- static int Rcpp::tzload (const char ∗name, struct state ∗const sp, const int doextend)
- static void Rcpp::gmtload (struct state ∗const sp)
- static struct tm ∗ Rcpp::gmtsub (const time_t ∗const timep, const int_fast32_t offset, struct tm ∗const tmp)
- attribute_hidden struct tm ∗ Rcpp::gmtime_ (const time_t ∗const x)

Variables

- static time_t const Rcpp::time_t_min = MINVAL(time_t, TYPE_BIT(time_t))
- static time_t const Rcpp::time_t_max = MAXVAL(time_t, TYPE_BIT(time_t))
- static const char Rcpp::gmt [ ] = "GMT"
- static const int Rcpp::mon_lengths [2][MONSPERYEAR]
- static const int Rcpp::year_lengths [2]
- static int Rcpp::gmt_is_set
- static struct state Rcpp::gmtmem
- static struct tm Rcpp::tm

7.398.1 Macro Definition Documentation

7.398.1.1 _NO_OLDNAMES

```
#define _NO_OLDNAMES /* avoid tznames */
```

Definition at line 120 of file date.cpp.

7.398.1.2 AVGSECSPERYEAR

```
#define AVGSECSPERYEAR 31556952L
```

Definition at line 145 of file date.cpp.

7.398.1.3 BIGGEST

```
#define BIGGEST(
              a,
              b ) (((a) > (b)) ? (a) : (b))
```

Definition at line 371 of file date.cpp.

7.398.1.4 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 28 of file date.cpp.

7.398.1.5 DAY_OF_YEAR

```
#define DAY_OF_YEAR 1 /* n - day of year */
```

Definition at line 417 of file date.cpp.

7.398.1.6 days_in_year

```
#define days_in_year(
             year ) (isleap(year) ?  366 :  365)
```

7.398.1.7 DAYSPERLYEAR

```
#define DAYSPERLYEAR 366
```

Definition at line 296 of file date.cpp.

7.398.1.8 DAYSPERNYEAR

```
#define DAYSPERNYEAR 365
```

Definition at line 295 of file date.cpp.

7.398.1.9 DAYSPERWEEK

```
#define DAYSPERWEEK 7
```

Definition at line 294 of file date.cpp.

7.398.1.10 EOVERFLOW

```
#define EOVERFLOW 79
```

Definition at line 126 of file date.cpp.

7.398.1.11 EPOCH_WDAY

```
#define EPOCH_WDAY TM_THURSDAY
```

Definition at line 325 of file date.cpp.

7.398.1.12 EPOCH_YEAR

```
#define EPOCH_YEAR 1970
```

Definition at line 324 of file date.cpp.

7.398.1.13 gmtptr

```
#define gmtptr (&gmtmem)
```

Definition at line 434 of file date.cpp.

7.398.1.14 GRANDPARENTED

```
#define GRANDPARENTED "Local time zone must be set--see zic manual page"
```

Definition at line 143 of file date.cpp.

7.398.1.15 HOURSPERDAY

```
#define HOURSPERDAY 24
```

Definition at line 293 of file date.cpp.

7.398.1.16 INITIALIZE

```
#define INITIALIZE(
              x ) (x = 0)
```

Definition at line 149 of file date.cpp.

7.398.1.17 is_digit

```
#define is_digit(
              c ) ((unsigned)(c) - '0' <= 9)
```

Definition at line 148 of file date.cpp.

7.398.1.18 isleap [1/2]

```
#define isleap(
              y ) ((((y) % 4) == 0 && ((y) % 100) != 0) || ((y) % 400) == 0)
```

Definition at line 327 of file date.cpp.

7.398.1.19 isleap [2/2]

```
#define isleap(
              y ) (((y) % 4) == 0 && (((y) % 100) != 0 || ((y) % 400) == 0))
```

Definition at line 327 of file date.cpp.

7.398.1.20 isleap_sum

```
#define isleap_sum(
              a,
              b ) isleap((a) % 400 + (b) % 400)
```

Definition at line 341 of file date.cpp.

7.398.1.21 JULIAN_DAY

```
#define JULIAN_DAY 0 /* Jn - Julian day */
```

Definition at line 416 of file date.cpp.

7.398.1.22 MAXVAL

```
#define MAXVAL(
              t,
              b )
```

Value:
```
((t) (((t) 1 « ((b) - 1 - TYPE_SIGNED(t)))                         \
     - 1 + ((t) 1 « ((b) - 1 - TYPE_SIGNED(t)))))
```

Definition at line 154 of file date.cpp.

7.398.1.23 MINSPERHOUR

```
#define MINSPERHOUR 60
```

Definition at line 292 of file date.cpp.

7.398.1.24 MINVAL

```
#define MINVAL(
              t,
              b )  ((t) (TYPE_SIGNED(t) ?  - TWOS_COMPLEMENT(t) - MAXVAL(t, b) :  0))
```

Definition at line 157 of file date.cpp.

7.398.1.25 MONSPERYEAR

```
#define MONSPERYEAR 12
```

Definition at line 299 of file date.cpp.

7.398.1.26 MONTH_NTH_DAY_OF_WEEK

```
#define MONTH_NTH_DAY_OF_WEEK 2 /* Mm.n.d - month, week, day of week */
```

Definition at line 418 of file date.cpp.

7.398.1.27 MY_TZNAME_MAX

```
#define MY_TZNAME_MAX 255
```

Definition at line 377 of file date.cpp.

7.398.1.28 OPEN_MODE

```
#define OPEN_MODE O_RDONLY
```

Definition at line 355 of file date.cpp.

7.398.1.29 SECSPERDAY

```
#define SECSPERDAY ((int_fast32_t) SECSPERHOUR * HOURSPERDAY)
```

Definition at line 298 of file date.cpp.

7.398.1.30 SECSPERHOUR

```
#define SECSPERHOUR (SECSPERMIN * MINSPERHOUR)
```

Definition at line 297 of file date.cpp.

7.398.1.31 SECSPERMIN

```
#define SECSPERMIN 60
```

Definition at line 291 of file date.cpp.

7.398.1.32 SECSPERREPEAT

```
#define SECSPERREPEAT ((int_fast64_t) YEARSPERREPEAT * (int_fast64_t) AVGSECSPERYEAR)
```

Definition at line 146 of file date.cpp.

7.398.1.33 SECSPERREPEAT_BITS

```
#define SECSPERREPEAT_BITS 34 /* ceil(log2(SECSPERREPEAT)) */
```

Definition at line 147 of file date.cpp.

7.398.1.34 TM_APRIL

```
#define TM_APRIL 3
```

Definition at line 312 of file date.cpp.

7.398.1.35 TM_AUGUST

```
#define TM_AUGUST 7
```

Definition at line 316 of file date.cpp.

7.398.1.36 TM_DECEMBER

```
#define TM_DECEMBER 11
```

Definition at line 320 of file date.cpp.

7.398.1.37 TM_FEBRUARY

```
#define TM_FEBRUARY 1
```

Definition at line 310 of file date.cpp.

7.398.1.38 TM_FRIDAY

```
#define TM_FRIDAY 5
```

Definition at line 306 of file date.cpp.

7.398.1.39 TM_JANUARY

```
#define TM_JANUARY 0
```

Definition at line 309 of file date.cpp.

7.398.1.40 TM_JULY

```
#define TM_JULY 6
```

Definition at line 315 of file date.cpp.

7.398.1.41 TM_JUNE

```
#define TM_JUNE 5
```

Definition at line 314 of file date.cpp.

7.398.1.42 TM_MARCH

```
#define TM_MARCH 2
```

Definition at line 311 of file date.cpp.

7.398.1.43 TM_MAY

```
#define TM_MAY 4
```

Definition at line 313 of file date.cpp.

7.398.1.44 TM_MONDAY

```
#define TM_MONDAY 1
```

Definition at line 302 of file date.cpp.

7.398.1.45 TM_NOVEMBER

```
#define TM_NOVEMBER 10
```

Definition at line 319 of file date.cpp.

7.398.1.46 TM_OCTOBER

```
#define TM_OCTOBER 9
```

Definition at line 318 of file date.cpp.

7.398.1.47 TM_SATURDAY

```
#define TM_SATURDAY 6
```

Definition at line 307 of file date.cpp.

7.398.1.48 TM_SEPTEMBER

```
#define TM_SEPTEMBER 8
```

Definition at line 317 of file date.cpp.

7.398.1.49 TM_SUNDAY

```
#define TM_SUNDAY 0
```

Definition at line 301 of file date.cpp.

7.398.1.50 TM_THURSDAY

```
#define TM_THURSDAY 4
```

Definition at line 305 of file date.cpp.

7.398.1.51 TM_TUESDAY

```
#define TM_TUESDAY 2
```

Definition at line 303 of file date.cpp.

7.398.1.52 TM_WEDNESDAY

```
#define TM_WEDNESDAY 3
```

Definition at line 304 of file date.cpp.

7.398.1.53 TM_YEAR_BASE

```
#define TM_YEAR_BASE 1900
```

Definition at line 322 of file date.cpp.

7.398.1.54 TWOS_COMPLEMENT

```
#define TWOS_COMPLEMENT(
             t ) ((t) ~ (t) 0 < 0)
```

Definition at line 142 of file date.cpp.

7.398.1.55 TYPE_BIT

```
#define TYPE_BIT(
             type ) (sizeof (type) * CHAR_BIT)
```

Definition at line 139 of file date.cpp.

7.398.1.56 TYPE_INTEGRAL

```
#define TYPE_INTEGRAL(
              type ) (((type) 0.5) != 0.5)
```

Definition at line 141 of file date.cpp.

7.398.1.57 TYPE_SIGNED

```
#define TYPE_SIGNED(
              type ) (((type) -1) < 0)
```

Definition at line 140 of file date.cpp.

7.398.1.58 TZ_MAGIC

```
#define TZ_MAGIC "TZif"
```

Definition at line 203 of file date.cpp.

7.398.1.59 TZ_MAX_CHARS

```
#define TZ_MAX_CHARS 100 /* Maximum number of abbreviation characters */
```

Definition at line 283 of file date.cpp.

7.398.1.60 TZ_MAX_LEAPS

```
#define TZ_MAX_LEAPS 50 /* Maximum number of leap second corrections */
```

Definition at line 288 of file date.cpp.

7.398.1.61 TZ_MAX_TIMES

```
#define TZ_MAX_TIMES 1200
```

Definition at line 265 of file date.cpp.

7.398.1.62 TZ_MAX_TYPES

```
#define TZ_MAX_TYPES 256 /* Limited by what (unsigned char)'s can hold */
```

Definition at line 270 of file date.cpp.

7.398.1.63 TZDEFAULT

```
#define TZDEFAULT "localtime"
```

Definition at line 192 of file date.cpp.

7.398.1.64 TZDEFRULES

```
#define TZDEFRULES "America/New_York"
```

Definition at line 196 of file date.cpp.

7.398.1.65 TZDEFRULESTRING

```
#define TZDEFRULESTRING ",M4.1.0,M10.5.0"
```

Definition at line 368 of file date.cpp.

7.398.1.66 TZDIR

```
#define TZDIR "/usr/local/etc/zoneinfo" /* Time zone object file directory */
```

Definition at line 188 of file date.cpp.

7.398.1.67 TZFILE_H

```
#define TZFILE_H
```

Definition at line 168 of file date.cpp.

7.398.1.68 YEARSPERREPEAT

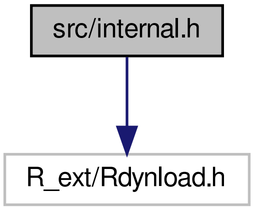
```
#define YEARSPERREPEAT 400 /* years before a Gregorian repeat */
```

Definition at line 144 of file date.cpp.

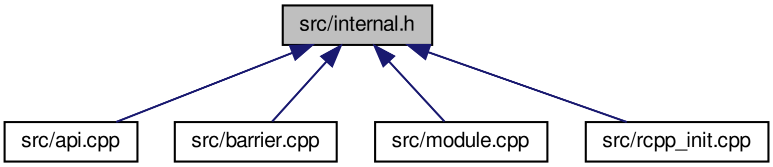
7.399 src/internal.h File Reference

```
#include <R_ext/Rdynload.h>
```

Include dependency graph for internal.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define CALLFUN_0(name) SEXP name()
- #define CALLFUN_1(name) SEXP name(SEXP)
- #define CALLFUN_2(name) SEXP name(SEXP,SEXP)
- #define CALLFUN_3(name) SEXP name(SEXP,SEXP,SEXP)
- #define CALLFUN_4(name) SEXP name(SEXP,SEXP,SEXP,SEXP)
- #define CALLFUN_5(name) SEXP name(SEXP,SEXP,SEXP,SEXP,SEXP)
- #define EXTFUN(name) SEXP name(SEXP)
- #define MAX_ARGS 65
- #define UNPACK_EXTERNAL_ARGS(__CARGS__, __P__)
- #define RCPP_FUN_1(__OUT__, __NAME__, ___0)
- #define RCPP_FUN_2(__OUT__, __NAME__, ___0, ___1)
- #define RCPP_FUN_3(__OUT__, __NAME__, ___0, ___1, ___2)
- #define RCPP_FUN_4(__OUT__, __NAME__, ___0, ___1, ___2, ___3)

Functions

- SEXP get_Rcpp_protection_stack ()
- CALLFUN_1 (as_character_externalptr)
- CALLFUN_1 (Class__name)
- CALLFUN_1 (Class__has_default_constructor)
- CALLFUN_1 (CppClass__complete)
- CALLFUN_1 (CppClass__methods)
- CALLFUN_1 (Module__classes_info)
- CALLFUN_1 (Module__complete)
- CALLFUN_1 (Module__functions_arity)
- CALLFUN_1 (Module__functions_names)
- CALLFUN_2 (Module__get_class)
- CALLFUN_2 (Module__has_class)
- CALLFUN_2 (Module__has_function)
- CALLFUN_2 (Module__get_function)
- CALLFUN_1 (Module__name)
- CALLFUN_2 (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)
- CALLFUN_0 (getRcppVersionStrings)
- 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 ∗)

7.399.1 Macro Definition Documentation

7.399.1.1 CALLFUN_0

```
#define CALLFUN_0(
              name ) SEXP name()
```

Definition at line 27 of file internal.h.

7.399.1.2 CALLFUN_1

```
#define CALLFUN_1(
              name ) SEXP name(SEXP)
```

Definition at line 28 of file internal.h.

7.399.1.3 CALLFUN_2

```
#define CALLFUN_2(
              name ) SEXP name(SEXP,SEXP)
```

Definition at line 29 of file internal.h.

7.399.1.4 CALLFUN_3

```
#define CALLFUN_3(
              name ) SEXP name(SEXP,SEXP,SEXP)
```

Definition at line 30 of file internal.h.

7.399.1.5 CALLFUN_4

```
#define CALLFUN_4(
              name ) SEXP name(SEXP,SEXP,SEXP,SEXP)
```

Definition at line 31 of file internal.h.

7.399.1.6 CALLFUN_5

```
#define CALLFUN_5(
                name ) SEXP name(SEXP,SEXP,SEXP,SEXP,SEXP)
```

Definition at line 32 of file internal.h.

7.399.1.7 EXTFUN

```
#define EXTFUN(
                name ) SEXP name(SEXP)
```

Definition at line 33 of file internal.h.

7.399.1.8 MAX_ARGS

```
#define MAX_ARGS 65
```

Definition at line 38 of file internal.h.

7.399.1.9 RCPP_FUN_1

```
#define RCPP_FUN_1(
                __OUT__,
                __NAME__,
                ___0 )
```

Value:

```
__OUT__ RCPP_DECORATE(__NAME__)(___0);           \
SEXP __NAME__(SEXP x0) {                         \
    SEXP res = R_NilValue;                       \
    BEGIN_RCPP                                   \
    res = ::Rcpp::wrap(RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0))); \
    return res;                                  \
    END_RCPP                                     \
}                                                \
__OUT__ RCPP_DECORATE(__NAME__)(___0)
```

Definition at line 48 of file internal.h.

7.399.1.10 RCPP_FUN_2

```
#define RCPP_FUN_2(
                __OUT__,
                __NAME__,
                ___0,
                ___1 )
```

Value:

```
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1);      \
SEXP __NAME__(SEXP x0, SEXP x1) {                 \
    SEXP res = R_NilValue;                        \
    BEGIN_RCPP                                    \
    res = ::Rcpp::wrap(RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0), \
                                               ::Rcpp::internal::converter(x1))); \
    return res;                                   \
    END_RCPP                                      \
}                                                 \
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1)
```

Definition at line 60 of file internal.h.

7.399.1.11 RCPP_FUN_3

```
#define RCPP_FUN_3(
                __OUT__,
                __NAME__,
                ___0,
                ___1,
                ___2 )
```

Value:

```
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1, ___2);      \
SEXP __NAME__(SEXP x0, SEXP x1, SEXP x2) {              \
    SEXP res = R_NilValue;                              \
    BEGIN_RCPP                                          \
    res = ::Rcpp::wrap(RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0),   \
                                               ::Rcpp::internal::converter(x1),   \
                                               ::Rcpp::internal::converter(x2))); \
    return res;                                         \
    END_RCPP                                            \
}                                                       \
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1, ___2)
```

Definition at line 72 of file internal.h.

7.399.1.12 RCPP_FUN_4

```
#define RCPP_FUN_4(
                __OUT__,
                __NAME__,
                ___0,
                ___1,
                ___2,
                ___3 )
```

Value:

```
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1, ___2, ___3);      \
SEXP __NAME__(SEXP x0, SEXP x1, SEXP x2, SEXP x3) {           \
    SEXP res = R_NilValue;                                    \
    BEGIN_RCPP                                                \
    res = ::Rcpp::wrap( RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0),   \
                                                ::Rcpp::internal::converter(x1),   \
                                                ::Rcpp::internal::converter(x2),   \
                                                ::Rcpp::internal::converter(x3))); \
    return res;                                               \
    END_RCPP                                                  \
}                                                             \
__OUT__ RCPP_DECORATE(__NAME__)(___0, ___1, ___2, ___3)
```

Definition at line 85 of file internal.h.

7.399.1.13 UNPACK_EXTERNAL_ARGS

```
#define UNPACK_EXTERNAL_ARGS(
                __CARGS__,
                __P__ )
```

Value:

```
SEXP __CARGS__[MAX_ARGS];                         \
int nargs = 0;                                    \
for (; nargs<MAX_ARGS; nargs++) {                 \
    if (Rf_isNull(__P__)) break;                  \
    __CARGS__[nargs] = CAR(__P__);                \
    __P__ = CDR(__P__);                           \
}
```

Definition at line 39 of file internal.h.

7.399.2 Function Documentation

7.399.2.1 CALLFUN_0() [1/6]

```
CALLFUN_0 (
              get_rcpp_cache   )
```

7.399.2.2 CALLFUN_0() [2/6]

```
CALLFUN_0 (
             getRcppVersionStrings  )
```

7.399.2.3 CALLFUN_0() [3/6]

```
CALLFUN_0 (
             init_Rcpp_cache  )
```

7.399.2.4 CALLFUN_0() [4/6]

```
CALLFUN_0 (
             rcpp_can_use_cxx0x  )
```

7.399.2.5 CALLFUN_0() [5/6]

```
CALLFUN_0 (
             rcpp_can_use_cxx11  )
```

7.399.2.6 CALLFUN_0() [6/6]

```
CALLFUN_0 (
             rcpp_capabilities  )
```

7.399.2.7 CALLFUN_1() [1/11]

```
CALLFUN_1 (
             as_character_externalptr  )
```

7.399.2.8 CALLFUN_1() [2/11]

```
CALLFUN_1 (
            Class__has_default_constructor  )
```

7.399.2.9 CALLFUN_1() [3/11]

```
CALLFUN_1 (
            Class__name  )
```

7.399.2.10 CALLFUN_1() [4/11]

```
CALLFUN_1 (
            CppClass__complete  )
```

7.399.2.11 CALLFUN_1() [5/11]

```
CALLFUN_1 (
            CppClass__methods  )
```

7.399.2.12 CALLFUN_1() [6/11]

```
CALLFUN_1 (
            Module__classes_info  )
```

7.399.2.13 CALLFUN_1() [7/11]

```
CALLFUN_1 (
            Module__complete  )
```

7.399.2.14 CALLFUN_1() [8/11]

```
CALLFUN_1 (
            Module__functions_arity )
```

7.399.2.15 CALLFUN_1() [9/11]

```
CALLFUN_1 (
            Module__functions_names )
```

7.399.2.16 CALLFUN_1() [10/11]

```
CALLFUN_1 (
            Module__name )
```

7.399.2.17 CALLFUN_1() [11/11]

```
CALLFUN_1 (
            rcpp_error_recorder )
```

7.399.2.18 CALLFUN_2() [1/5]

```
CALLFUN_2 (
            CppObject__finalize )
```

7.399.2.19 CALLFUN_2() [2/5]

```
CALLFUN_2 (
            Module__get_class )
```

7.399.2.20 CALLFUN_2() [3/5]

```
CALLFUN_2 (
             Module__get_function )
```

7.399.2.21 CALLFUN_2() [4/5]

```
CALLFUN_2 (
             Module__has_class )
```

7.399.2.22 CALLFUN_2() [5/5]

```
CALLFUN_2 (
             Module__has_function )
```

7.399.2.23 CALLFUN_3()

```
CALLFUN_3 (
             CppField__get )
```

7.399.2.24 CALLFUN_4()

```
CALLFUN_4 (
             CppField__set )
```

7.399.2.25 EXTFUN() [1/7]

```
EXTFUN (
             class__dummyInstance )
```

7.399.2.26 EXTFUN() [2/7]

```
EXTFUN (
            class__newInstance )
```

7.399.2.27 EXTFUN() [3/7]

```
EXTFUN (
            CppMethod__invoke )
```

7.399.2.28 EXTFUN() [4/7]

```
EXTFUN (
            CppMethod__invoke_notvoid )
```

7.399.2.29 EXTFUN() [5/7]

```
EXTFUN (
            CppMethod__invoke_void )
```

7.399.2.30 EXTFUN() [6/7]

```
EXTFUN (
            InternalFunction_invoke )
```

7.399.2.31 EXTFUN() [7/7]

```
EXTFUN (
            Module__invoke )
```

7.399.2.32 get_Rcpp_protection_stack()

```
SEXP get_Rcpp_protection_stack ( )
```

7.399.2.33 init_Rcpp_routines()

```
void init_Rcpp_routines (
              DllInfo * info )
```

Definition at line 78 of file rcpp_init.cpp.

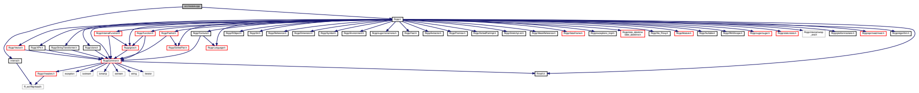
References callEntries, and extEntries.

Referenced by R_init_Rcpp().

7.400 src/module.cpp File Reference

```
#include <Rcpp.h>
#include "internal.h"
```

Include dependency graph for module.cpp:



Namespaces

- Rcpp

  Rcpp API.

Macros

- #define COMPILING_RCPP
- #define CHECK_DUMMY_OBJ(p) if (p == rcpp_dummy_pointer) throw Rcpp::not_initialized()

Typedefs

- typedef Rcpp::XPtr< Rcpp::Module > XP_Module
- typedef Rcpp::XPtr< Rcpp::class_Base > XP_Class
- typedef Rcpp::XPtr< Rcpp::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

7.400.1 Macro Definition Documentation

7.400.1.1 CHECK_DUMMY_OBJ

```
#define CHECK_DUMMY_OBJ(
              p ) if (p == rcpp_dummy_pointer) throw Rcpp::not_initialized()
```

Definition at line 149 of file module.cpp.

7.400.1.2 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file module.cpp.

7.400.2 Typedef Documentation

7.400.2.1 XP_Class

```
typedef Rcpp::XPtr<Rcpp::class_Base> XP_Class
```

Definition at line 28 of file module.cpp.

7.400.2.2 XP_Function

```
typedef Rcpp::XPtr<Rcpp::CppFunctionBase> XP_Function
```

Definition at line 29 of file module.cpp.

7.400.2.3 XP_Module

```
typedef Rcpp::XPtr<Rcpp::Module> XP_Module
```

Definition at line 27 of file module.cpp.

7.400.3 Function Documentation

7.400.3.1 class__dummyInstance()

```
SEXP class__dummyInstance (
             SEXP args )
```

Definition at line 152 of file module.cpp.

References rcpp_dummy_pointer.

7.400.3.2 class__newInstance()

```
SEXP class__newInstance (
             SEXP args )
```

Definition at line 137 of file module.cpp.

References UNPACK_EXTERNAL_ARGS.

7.400.3.3 CppMethod__invoke()

```
SEXP CppMethod__invoke (
             SEXP args )
```

Definition at line 166 of file module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.400.3.4 CppMethod__invoke_notvoid()

```
SEXP CppMethod__invoke_notvoid (
             SEXP args )
```

Definition at line 204 of file module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.400.3.5 CppMethod__invoke_void()

```
SEXP CppMethod__invoke_void (
              SEXP args )
```

Definition at line 185 of file module.cpp.

References CHECK_DUMMY_OBJ, and UNPACK_EXTERNAL_ARGS.

7.400.3.6 getCurrentScope()

```
Rcpp::Module* getCurrentScope ( )
```

Definition at line 227 of file module.cpp.

References Rcpp::current_scope.

7.400.3.7 InternalFunction_invoke()

```
SEXP InternalFunction_invoke (
              SEXP args )
```

Definition at line 117 of file module.cpp.

References BEGIN_RCPP, END_RCPP, and UNPACK_EXTERNAL_ARGS.

7.400.3.8 Module__invoke()

```
SEXP Module__invoke (
              SEXP args )
```

Definition at line 126 of file module.cpp.

References BEGIN_RCPP, END_RCPP, and UNPACK_EXTERNAL_ARGS.

7.400.3.9 RCPP_FUN_1() [1/14]

```
RCPP_FUN_1 (
             bool ,
             Class__has_default_constructor ,
             XP_Class cl )
```

Definition at line 31 of file module.cpp.

7.400.3.10 RCPP_FUN_1() [2/14]

```
RCPP_FUN_1 (
             bool ,
             CppObject__needs_init ,
             SEXP xp )
```

Definition at line 55 of file module.cpp.

7.400.3.11 RCPP_FUN_1() [3/14]

```
RCPP_FUN_1 (
             Rcpp::CharacterVector ,
             CppClass__complete ,
             XP_Class cl )
```

Definition at line 98 of file module.cpp.

7.400.3.12 RCPP_FUN_1() [4/14]

```
RCPP_FUN_1 (
             Rcpp::CharacterVector ,
             CppClass__methods ,
             XP_Class cl )
```

Definition at line 58 of file module.cpp.

7.400.3.13 RCPP_FUN_1() [5/14]

```
RCPP_FUN_1 (
             Rcpp::CharacterVector ,
             CppClass__properties ,
             XP_Class cl )
```

Definition at line 61 of file module.cpp.

7.400.3.14 RCPP_FUN_1() [6/14]

```
RCPP_FUN_1 (
             Rcpp::CharacterVector ,
             Module__complete ,
             XP_Module module )
```

Definition at line 95 of file module.cpp.

7.400.3.15 RCPP_FUN_1() [7/14]

```
RCPP_FUN_1 (
             Rcpp::CharacterVector ,
             Module__functions_names ,
             XP_Module module )
```

Definition at line 86 of file module.cpp.

7.400.3.16 RCPP_FUN_1() [8/14]

```
RCPP_FUN_1 (
             Rcpp::IntegerVector ,
             CppClass__methods_arity ,
             XP_Class cl )
```

Definition at line 68 of file module.cpp.

7.400.3.17 RCPP_FUN_1() [9/14]

```
RCPP_FUN_1 (
             Rcpp::IntegerVector ,
             Module__functions_arity ,
             XP_Module module )
```

Definition at line 83 of file module.cpp.

7.400.3.18 RCPP_FUN_1() [10/14]

```
RCPP_FUN_1 (
             Rcpp::List ,
             CppClass__property_classes ,
             XP_Class cl )
```

Definition at line 64 of file module.cpp.

7.400.3.19 RCPP_FUN_1() [11/14]

```
RCPP_FUN_1 (
             Rcpp::List ,
             Module__classes_info ,
             XP_Module module )
```

Definition at line 92 of file module.cpp.

7.400.3.20 RCPP_FUN_1() [12/14]

```
RCPP_FUN_1 (
             Rcpp::LogicalVector ,
             CppClass__methods_voidness ,
             XP_Class cl )
```

Definition at line 71 of file module.cpp.

7.400.3.21 RCPP_FUN_1() [13/14]

```
RCPP_FUN_1 (
             std::string ,
             Class__name ,
             XP_Class cl )
```

Definition at line 43 of file module.cpp.

7.400.3.22 RCPP_FUN_1() [14/14]

```
RCPP_FUN_1 (
             std::string ,
             Module__name ,
             XP_Module module )
```

Definition at line 89 of file module.cpp.

7.400.3.23 RCPP_FUN_2() [1/9]

```
RCPP_FUN_2 (
             bool ,
             Class__has_method ,
             XP_Class cl,
             std::string m )
```

Definition at line 37 of file module.cpp.

7.400.3.24 RCPP_FUN_2() [2/9]

```
RCPP_FUN_2 (
             bool ,
             Class__has_property ,
             XP_Class cl,
             std::string m )
```

Definition at line 40 of file module.cpp.

7.400.3.25 RCPP_FUN_2() [3/9]

```
RCPP_FUN_2 (
             bool ,
             CppClass__property_is_readonly ,
             XP_Class cl,
             std::string p )
```

Definition at line 76 of file module.cpp.

7.400.3.26 RCPP_FUN_2() [4/9]

```
RCPP_FUN_2 (
             bool ,
             Module__has_class ,
             XP_Module module,
             std::string cl )
```

Definition at line 49 of file module.cpp.

7.400.3.27 RCPP_FUN_2() [5/9]

```
RCPP_FUN_2 (
             bool ,
             Module__has_function ,
             XP_Module module,
             std::string met )
```

Definition at line 46 of file module.cpp.

7.400.3.28 RCPP_FUN_2() [6/9]

```
RCPP_FUN_2 (
             Rcpp::CppClass ,
             Module__get_class ,
             XP_Module module,
             std::string cl )
```

Definition at line 52 of file module.cpp.

7.400.3.29 RCPP_FUN_2() [7/9]

```
RCPP_FUN_2 (
             SEXP ,
             CppObject__finalize ,
             XP_Class cl,
             SEXP obj )
```

Definition at line 111 of file module.cpp.

7.400.3.30 RCPP_FUN_2() [8/9]

```
RCPP_FUN_2 (
             SEXP ,
             Module__get_function ,
             XP_Module module,
             std::string fun )
```

Definition at line 34 of file module.cpp.

7.400.3.31 RCPP_FUN_2() [9/9]

```
RCPP_FUN_2 (
             std::string ,
             CppClass__property_class ,
             XP_Class cl,
             std::string p )
```

Definition at line 79 of file module.cpp.

7.400.3.32 RCPP_FUN_3()

```
RCPP_FUN_3 (
             SEXP ,
             CppField__get ,
             XP_Class cl,
             SEXP field_xp,
             SEXP obj )
```

Definition at line 104 of file module.cpp.

7.400.3.33 RCPP_FUN_4()

```
RCPP_FUN_4 (
             SEXP ,
             CppField__set ,
             XP_Class cl,
             SEXP field_xp,
             SEXP obj,
             SEXP value )
```

Definition at line 107 of file module.cpp.

7.400.3.34 setCurrentScope()

```
void setCurrentScope (
             Rcpp::Module * scope )
```

Definition at line 230 of file module.cpp.

References Rcpp::current_scope.

7.400.4 Variable Documentation

7.400.4.1 rcpp_dummy_pointer

```
SEXP rcpp_dummy_pointer = R_NilValue
```

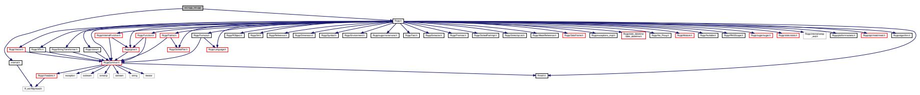
Definition at line 147 of file module.cpp.

Referenced by class__dummyInstance().

7.401 src/rcpp_init.cpp File Reference

```
#include <Rcpp.h>
#include "internal.h"
```

Include dependency graph for rcpp_init.cpp:



Macros

- #define COMPILING_RCPP
- #define CALLDEF(name, n) {#name, (DL_FUNC) &name, n}
- #define EXTDEF(name) {#name, (DL_FUNC) &name, -1}
- #define RCPP_REGISTER(__FUN__) R_RegisterCCallable("Rcpp", #__FUN__ , (DL_FUNC)__FUN__);

Functions

- void init_Rcpp_routines (DllInfo ∗info)
- void registerFunctions ()
- void R_unload_Rcpp (DllInfo ∗)
- void R_init_Rcpp (DllInfo ∗dllinfo)

Variables

- static R_CallMethodDef callEntries [ ]
- static R_ExternalMethodDef extEntries [ ]

7.401.1 Macro Definition Documentation

7.401.1.1 CALLDEF

```
#define CALLDEF(
              name,
              n ) {#name, (DL_FUNC) &name, n}
```

Definition at line 28 of file rcpp_init.cpp.

7.401.1.2 COMPILING_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file rcpp_init.cpp.

7.401.1.3 EXTDEF

```
#define EXTDEF(
             name ) {#name, (DL_FUNC) &name, -1}
```

Definition at line 29 of file rcpp_init.cpp.

7.401.1.4 RCPP_REGISTER

```
#define RCPP_REGISTER(
             __FUN__ ) R_RegisterCCallable( "Rcpp", #__FUN__ , (DL_FUNC)__FUN__ );
```

7.401.2 Function Documentation

7.401.2.1 init_Rcpp_routines()

```
void init_Rcpp_routines (
             DllInfo * info )
```

Definition at line 78 of file rcpp_init.cpp.

References callEntries, and extEntries.

Referenced by R_init_Rcpp().

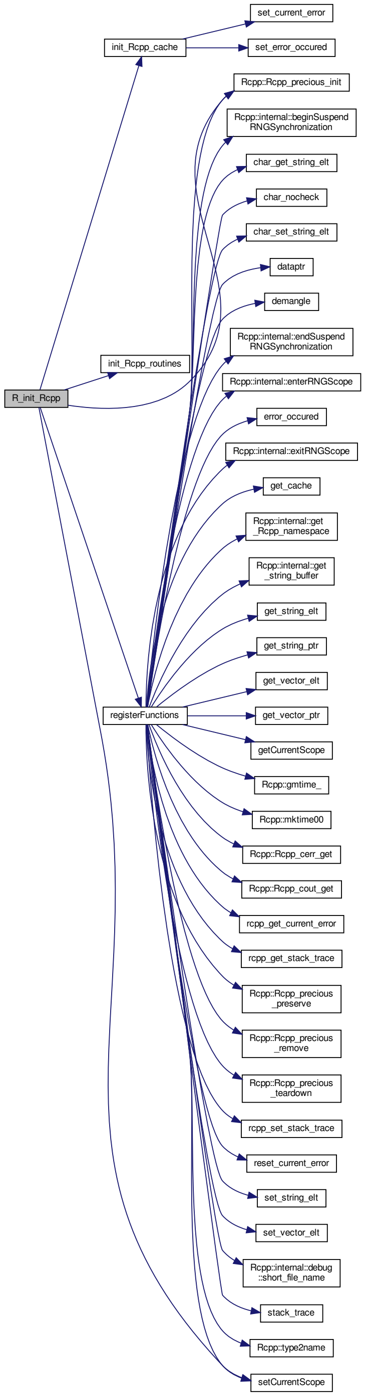
7.401.2.2 R_init_Rcpp()

```
void R_init_Rcpp (
             DllInfo * dllinfo )
```

Definition at line 137 of file rcpp_init.cpp.

References init_Rcpp_cache(), init_Rcpp_routines(), Rcpp::Rcpp_precious_init(), registerFunctions(), and setCurrent←
Scope().

Here is the call graph for this function:



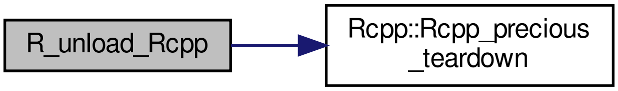
7.401.2.3 R_unload_Rcpp()

```
void R_unload_Rcpp (
            DllInfo *  )
```

Definition at line 133 of file rcpp_init.cpp.

References Rcpp::Rcpp_precious_teardown().

Here is the call graph for this function:



7.401.2.4 registerFunctions()

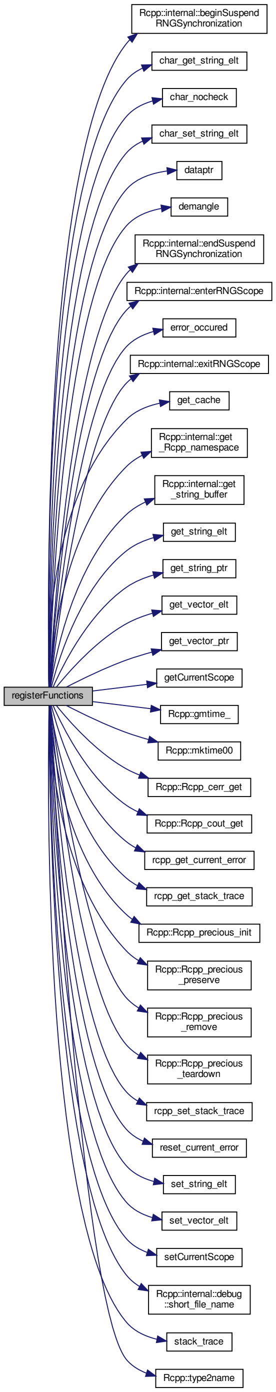
```
void registerFunctions ( )
```

Definition at line 88 of file rcpp_init.cpp.

References Rcpp::internal::beginSuspendRNGSynchronization(), char_get_string_elt(), char_nocheck(), char_set_string_elt(), dataptr(), demangle(), Rcpp::internal::endSuspendRNGSynchronization(), Rcpp::internal::enterRNGScope(), error_occured(), Rcpp::internal::exitRNGScope(), get_cache(), Rcpp::internal::get_Rcpp_namespace(), Rcpp::internal::get_string_buffer(), get_string_elt(), get_string_ptr(), get_vector_elt(), get_vector_ptr(), getCurrentScope(), Rcpp::gmtime_(), Rcpp::mktime00(), Rcpp::Rcpp_cerr_get(), Rcpp::Rcpp_cout_get(), rcpp_get_current_error(), rcpp_get_stack_trace(), Rcpp::Rcpp_precious_init(), Rcpp::Rcpp_precious_preserve(), Rcpp::Rcpp_precious_remove(), Rcpp::Rcpp_precious_teardown(), RCPP_REGISTER, rcpp_set_stack_trace(), reset_current_error(), set_string_elt(), set_vector_elt(), setCurrentScope(), Rcpp::internal::debug::short_file_name(), stack_trace(), and Rcpp::type2name().

Referenced by R_init_Rcpp().

Here is the call graph for this function:



7.401.3 Variable Documentation

7.401.3.1 callEntries

```
R_CallMethodDef callEntries[ ]  [static]
```

Definition at line 31 of file rcpp_init.cpp.

Referenced by init_Rcpp_routines(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

7.401.3.2 extEntries

```
R_ExternalMethodDef extEntries[ ]  [static]
```

Initial value:

```
= {
    EXTDEF(CppMethod__invoke),
    EXTDEF(CppMethod__invoke_void),
    EXTDEF(CppMethod__invoke_notvoid),
    EXTDEF(InternalFunction_invoke),
    EXTDEF(Module__invoke),
    EXTDEF(class__newInstance),
    EXTDEF(class__dummyInstance),
    {NULL, NULL, 0}
}
```

Definition at line 65 of file rcpp_init.cpp.

Referenced by init_Rcpp_routines().

Chapter 8

Example Documentation

8.1 functionCallback/RcppFunctionCallExample.cpp

An example providing a class RVectorFuncCall derived from class RcppFunction which implements a vector-values function transformVector(); the class gets initialized with an R function vecfunc passed along using Rcpp.

8.2 functionCallback/ExampleRCode.R

This files defines the R function vecfunc passed down to C++ and called via the wrapper function built using the sub-class of RcppFunction to call it.

8.3 RcppInline/external_pointer.r

A simple example (using inline) of external Pointer use via Rcpp::XPtr

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010     Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
require(Rcpp)
require(inline)
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
funx_old <- cxxfunction(signature(), '
        /* creating a pointer to a vector<int> */
        std::vector<int>* v = new std::vector<int> ;
        v->push_back( 1 ) ;
        v->push_back( 2 ) ;

        /* wrap the pointer as an external pointer */
        /* this automatically protected the external pointer from R garbage
           collection until p goes out of scope. */
        Rcpp::XPtr< std::vector<int> > p(v) ;

        /* return it back to R, since p goes out of scope after the return
           the external pointer is no more protected by p, but it gets
           protected by being on the R side */
        return( p ) ;
', plugin = "Rcpp" )
xp <- funx_old()
stopifnot( identical( typeof( xp ), "externalptr" ) )
# passing the pointer back to C++
funx_old <- cxxfunction(signature(x = "externalptr" ), '
        /* wrapping x as smart external pointer */
        /* The SEXP based constructor does not protect the SEXP from
           garbage collection automatically, it is already protected
           because it comes from the R side, however if you want to keep
           the Rcpp::XPtr object on the C(++) side
           and return something else to R, you need to protect the external
           pointer, by using the protect member function */
        Rcpp::XPtr< std::vector<int> > p(x) ;

        /* just return the front of the vector as a SEXP */
        return( Rcpp::wrap( p->front() ) ) ;
', plugin = "Rcpp" )
front <- funx_old(xp)
stopifnot( identical( front, 1L ) )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
## Use of the cppFunction() gives the ability to immediately compile embedded
## C++ directly within R without having to worry about header specification or
## Rcpp attributes.
cppFunction('
Rcpp::XPtr< std::vector<int> >  funx(){
    /* creating a pointer to a vector<int> */
    std::vector<int>* v = new std::vector<int> ;
    v->push_back( 1 ) ;
    v->push_back( 2 ) ;

    /* wrap the pointer as an external pointer */
    /* this automatically protected the external pointer from R garbage
     *   collection until p goes out of scope.
     */
    Rcpp::XPtr< std::vector<int> > p(v) ;

    /* return it back to R, since p goes out of scope after the return
     * the external pointer is no more protected by p, but it gets
     * protected by being on the R side
     */
    return( p ) ;
}')
xp <- funx()
stopifnot( identical( typeof( xp ), "externalptr" ) )
# passing the pointer back to C++
cppFunction('
SEXP funx_pt(Rcpp::XPtr< std::vector<int> > p){
    /* Wrapping x as smart external pointer */
    /* The SEXP based constructor does not protect the SEXP from
     * garbage collection automatically, it is already protected
     * because it comes from the R side, however if you want to keep
     * the Rcpp::XPtr object on the C(++) side
     * and return something else to R, you need to protect the external
     * pointer, by using the protect member function
     */
    /* Just return the front of the vector as a SEXP */
    return Rcpp::wrap(p->front());
}')
front <- funx_pt(xp)
stopifnot( identical( front, 1L ) )
```

8.4 RcppInline/RcppInlineExample.r

The 'distribution of determinant' example as a demonstration of how to use inline and Rcpp for a simple function.

```
#!/usr/bin/env r
suppressMessages(library(Rcpp))
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
suppressMessages(library(inline))
foo <- ’
    IntegerVector vec(10000);     // vec parameter viewed as vector of ints.
    int i = 0;
    for (int a = 0; a < 9; a++)
      for (int b = 0; b < 9; b++)
        for (int c = 0; c < 9; c++)
          for (int d = 0; d < 9; d++)
            vec(i++) = a*b - c*d;
    return vec;
’
funx_old <- cxxfunction(signature(), foo, plugin = "Rcpp" )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
cppFunction(’IntegerVector funx(){
    IntegerVector vec(10000);     // vec parameter viewed as vector of ints.
    int i = 0;
    for (int a = 0; a < 9; a++)
      for (int b = 0; b < 9; b++)
        for (int c = 0; c < 9; c++)
          for (int d = 0; d < 9; d++)
            vec(i++) = a*b - c*d;
    return vec;
}’)
dd.inline.rcpp <- function() {
    res <- funx()
    tabulate(res)
}
print(mean(replicate(100,system.time(dd.inline.rcpp())["elapsed"]),trim=0.05))
```

8.5 RcppInline/RcppInlineWithLibsExamples.r

Four simple examples of how to combine Rcpp and an external library (where we use the GNU GSL) using inline.

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2016  Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
suppressMessages(library(Rcpp))
suppressMessages(library(RcppGSL))
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
suppressMessages(library(inline))
firstExample_old <- function() {
    ## a really simple C program calling three functions from the GSL
    gslrng <- ’
    gsl_rng *r;
    gsl_rng_env_setup();
    double v;
    r = gsl_rng_alloc (gsl_rng_default);
    printf("  generator type: %s\\n", gsl_rng_name (r));
    printf("  seed = %lu\\n", gsl_rng_default_seed);
    v = gsl_rng_get (r);
    printf("  first value = %.0f\\n", v);
    gsl_rng_free(r);
    return R_NilValue;
    ’
    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    funx_old <- cxxfunction(signature(), gslrng,
                            includes="#include <gsl/gsl_rng.h>",
                            plugin="RcppGSL")
    cat("Calling first example\n")
    funx_old()
    invisible(NULL)
}
secondExample_old <- function() {
    ## now use Rcpp to pass down a parameter for the seed
    gslrng <- ’
    int seed = Rcpp::as<int>(par) ;
    gsl_rng *r;
    gsl_rng_env_setup();
    double v;
    r = gsl_rng_alloc (gsl_rng_default);
    gsl_rng_set (r, (unsigned long) seed);
    v = gsl_rng_get (r);
    #ifndef BeSilent
    printf("  generator type: %s\\n", gsl_rng_name (r));
    printf("  seed = %d\\n", seed);
    printf("  first value = %.0f\\n", v);
    #endif
    gsl_rng_free(r);
    return Rcpp::wrap(v) ;
    ’
    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output
    funx_old <- cxxfunction(signature(par="numeric"), gslrng,
                            includes="#include <gsl/gsl_rng.h>",
                            plugin="RcppGSL")
    cat("\n\nCalling second example without -DBeSilent set\n")
    print(funx_old(0))
    ## now override settings to add -D flag
    settings <- getPlugin("RcppGSL")
    settings$env$PKG_CPPFLAGS <- paste(settings$PKG_CPPFLAGS, "-DBeSilent")

    funx_old <- cxxfunction(signature(par="numeric"), gslrng,
                            includes="#include <gsl/gsl_rng.h>",
                            settings=settings)
    cat("\n\nCalling second example with -DBeSilent set\n")
    print(funx_old(0))
    invisible(NULL)
}
thirdExample_old <- function() {
    ## now use Rcpp to pass down a parameter for the seed, and a vector size
    gslrng <- ’
    int seed = Rcpp::as<int>(s) ;
    int len = Rcpp::as<int>(n);
    gsl_rng *r;
    gsl_rng_env_setup();
    std::vector<double> v(len);
    r = gsl_rng_alloc (gsl_rng_default);
    gsl_rng_set (r, (unsigned long) seed);
    for (int i=0; i<len; i++) {
       v[i] = gsl_rng_get (r);
    }
    gsl_rng_free(r);
    return Rcpp::wrap(v) ;
    ’
    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output
    funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
                            gslrng,
                            includes="#include <gsl/gsl_rng.h>",
                            plugin="RcppGSL")
    cat("\n\nCalling third example with seed and length\n")
    print(funx_old(0, 5))
    invisible(NULL)
}
fourthExample_old <- function() {
    ## now use Rcpp to pass down a parameter for the seed, and a vector size
    gslrng <- ’
    int seed = Rcpp::as<int>(s);
    int len = Rcpp::as<int>(n);
    gsl_rng *r;
    gsl_rng_env_setup();
    std::vector<double> v(len);
    r = gsl_rng_alloc (gsl_rng_default);
    gsl_rng_set (r, (unsigned long) seed);
    for (int i=0; i<len; i++) {
       v[i] = gsl_rng_get (r);
    }
    gsl_rng_free(r);
    return wrap(v);
    ’
    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output
    funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
                            gslrng,
                            includes=c("#include <gsl/gsl_rng.h>",
                                       "using namespace Rcpp;",
                                       "using namespace std;"),
                            plugin="RcppGSL")
    cat("\n\nCalling fourth example with seed, length and namespaces\n")
    print(funx_old(0, 5))
    invisible(NULL)
}
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
firstExample <- function() {
    ## a really simple C program calling three functions from the GSL
    sourceCpp(code=’
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
// [[Rcpp::depends(RcppGSL)]]
// [[Rcpp::export]]
SEXP funx(){
    gsl_rng *r;
    gsl_rng_env_setup();
    double v;

    r = gsl_rng_alloc (gsl_rng_default);

    printf("  generator type: %s\\n", gsl_rng_name (r));
    printf("  seed = %lu\\n", gsl_rng_default_seed);
    v = gsl_rng_get (r);
    printf("  first value = %.0f\\n", v);

    gsl_rng_free(r);
    return R_NilValue;
}’)

    cat("Calling first example\n")
    funx()
    invisible(NULL)
}
secondExample <- function() {

    ## now use Rcpp to pass down a parameter for the seed

    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output
    gslrng <- ’
    #include <RcppGSL.h>
    #include <gsl/gsl_rng.h>

    // [[Rcpp::depends(RcppGSL)]]

    // [[Rcpp::export]]
    double funx(int seed){

    gsl_rng *r;
    gsl_rng_env_setup();
    double v;

    r = gsl_rng_alloc (gsl_rng_default);

    gsl_rng_set (r, (unsigned long) seed);
    v = gsl_rng_get (r);
    #ifndef BeSilent
    printf("  generator type: %s\\n", gsl_rng_name (r));
    printf("  seed = %d\\n", seed);
    printf("  first value = %.0f\\n", v);
    #endif

    gsl_rng_free(r);
    return v;
    }'
    sourceCpp(code=gslrng, rebuild = TRUE)

    cat("\n\nCalling second example without -DBeSilent set\n")
    print(funx(0))


    ## now override settings to add -D flag
    o = Sys.getenv("PKG_CPPFLAGS")
    Sys.setenv("PKG_CPPFLAGS" = paste(o, "-DBeSilent"))

    sourceCpp(code=gslrng, rebuild = TRUE)

    # Restore environment flags
    Sys.setenv("PKG_CPPFLAGS" = o )

    cat("\n\nCalling second example with -DBeSilent set\n")
    print(funx(0))

    invisible(NULL)
}
thirdExample <- function() {

    ## now use Rcpp to pass down a parameter for the seed, and a vector size

    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output

    sourceCpp(code='
    #include <RcppGSL.h>
    #include <gsl/gsl_rng.h>

    // [[Rcpp::depends(RcppGSL)]]

    // [[Rcpp::export]]
    std::vector<double> funx(int seed, int len){

    gsl_rng *r;
    gsl_rng_env_setup();
    std::vector<double> v(len);

    r = gsl_rng_alloc (gsl_rng_default);

    gsl_rng_set (r, (unsigned long) seed);
    for (int i=0; i<len; i++) {
       v[i] = gsl_rng_get (r);
    }
    gsl_rng_free(r);

    return v;
    }')

    cat("\n\nCalling third example with seed and length\n")
    print(funx(0, 5))

    invisible(NULL)
}
fourthExample <- function() {

    ## now use Rcpp to pass down a parameter for the seed, and a vector size

    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output

    sourceCpp(code='
    #include <RcppGSL.h>
    #include <gsl/gsl_rng.h>
    using namespace Rcpp;
    using namespace std;

    // [[Rcpp::depends(RcppGSL)]]
    // [[Rcpp::export]]
    std::vector<double> funx(int seed, int len){
    gsl_rng *r;
    gsl_rng_env_setup();
    std::vector<double> v(len);

    r = gsl_rng_alloc (gsl_rng_default);

    gsl_rng_set (r, (unsigned long) seed);
    for (int i=0; i<len; i++) {
    v[i] = gsl_rng_get (r);
    }
    gsl_rng_free(r);

    return v;
    }’)

    cat("\n\nCalling fourth example with seed, length and namespaces\n")
    print(funx(0, 5))

    invisible(NULL)
}
firstExample()
secondExample()
thirdExample()
fourthExample()
```

8.6 RcppInline/RObject.r

A number of examples on how to use wrap() to automatically convert data types.

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010  Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
## NB This file is mostly historic and predates the unit tests.
##    Yet it still provides useful examples -- but the unitTests/
##    for vastly larger coverage
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
cat("===Doubles\n")
foo <- ’
        double d = Rcpp::as<double>(x);
        std::cout « "Returning twice the value of " « d « " : ";
        return(Rcpp::wrap( 2*d ) );
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x=2.123), "\n")
cat(funx(x=2), "\n")
##funx(x=’2’)  ## throws as expected
##funx(x=2:3)  ## throws as expected
cat("\n===Int\n")
foo <- ’
        int i = Rcpp::as<int>(x);
        std::cout « "Returning twice the value of " « i « " : ";
        return(Rcpp::wrap( 2*i ) );
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x=2), "\n")
cat(funx(x=2.2), "\n")
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
```

```
cat(funx(x=as.raw(2)), "\n")
cat("\n===String\n")
foo <- ’
        std::string s = Rcpp::as<std::string>(x);
        std::cout « "Returning twice the value of " « s « " : ";
        return(Rcpp::wrap( s+s ) );
        ’
funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x="abc"), "\n")
cat("\n===Raw (bytes)\n")
foo <- ’
        Rbyte i = Rcpp::as<Rbyte>(x) ;
        std::cout « "Returning twice the value of " « (int)i « " : ";
        return(Rcpp::wrap( (Rbyte)(2*i) ) );
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=2), "\n")
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=2L), "\n")
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=as.raw(2)), "\n")
cat("\n=== logical \n")
foo <- ’
bool b = Rcpp::as<bool>(x);
std::cout « "flip  " « ( b ? "TRUE" : "FALSE" ) « " : ";
return(Rcpp::wrap( !b ));
’
funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=TRUE) , "\n")  ; stopifnot( !res )
cat( res <- funx(x=FALSE), "\n" ) ; stopifnot( res)
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=2)  , "\n")   ; stopifnot( !res )
cat( res <- funx(x=0.0), "\n")   ; stopifnot( res)
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=2L), "\n")    ; stopifnot( !res )
cat( res <- funx(x=0L), "\n")    ; stopifnot( res)
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=as.raw(2)), "\n") ; stopifnot( !res )
cat( res <- funx(x=as.raw(0)), "\n") ; stopifnot( res)
### vectors
cat("\n===Int Vector via wrap\n")
foo <- ’
        std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
        std::cout « "Returning twice the value of vector : ";
        for (size_t i=0; i<iv.size(); i++) {
            iv[i] = 2*iv[i];
        }
        return(Rcpp::wrap(iv));
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Int Vector\n")
foo <- ’
        std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
        std::cout « "Returning twice the value of vector : ";
        for (size_t i=0; i<iv.size(); i++) {
            iv[i] = 2*iv[i];
        }
        return(Rcpp::wrap( iv ) );
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5+.1))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Double Vector\n")
foo <- ’
        std::vector<double> iv = Rcpp::as< std::vector<double> >(x) ;
        std::cout « "Returning twice the value of vector : ";
        for (size_t i=0; i<iv.size(); i++) {
            iv[i] = 2*iv[i];
        }
        return(Rcpp::wrap( iv ));
        ’
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0.1+2:5))
```

```
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Raw Vector\n")
foo <- '
        std::vector<Rbyte> iv = Rcpp::as< std::vector<Rbyte> >(x) ;
        std::cout « "Returning twice the value of vector : ";
        for (size_t i=0; i<iv.size(); i++) {
            iv[i] = 2*iv[i];
        }
        return(Rcpp::wrap( iv ));
        '
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(0:9)))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9))
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9+.1))
cat("\n=== vector<bool>\n")
foo <- '
std::vector<bool> bv = Rcpp::as< std::vector<bool> >(x) ;
std::cout « "Flip the value of vector : ";
for (size_t i=0; i<bv.size(); i++) {
    bv[i].flip() ;
}
return(Rcpp::wrap( bv ));
'
funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=c(TRUE,FALSE)))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(0:9)))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9))
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.numeric(0:9)))
cat("\n===String Vector\n")
foo <- '
        std::vector<std::string> iv = Rcpp::as< std::vector<std::string> >(x);
        std::cout « "Returning twice the value of vector : ";
        for (size_t i=0; i<iv.size(); i++) {
            iv[i] = iv[i] + iv[i];
        }
        return(Rcpp::wrap( iv ));
        '
funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=c("foo", "bar")))
### using std::set
cat("\n=== set<int>\n")
foo <- '
std::set<int> iv ;
iv.insert( 0 ) ;
iv.insert( 1 ) ;
iv.insert( 0 ) ;
return Rcpp::wrap( iv );'
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>" )
print(res <- funx())
stopifnot( identical( res, 0:1 ) )
cat("\n=== set<double>\n")
foo <- '
std::set<double> ds;
ds.insert( 0.0 );
ds.insert( 1.0 );
ds.insert( 0.0 );
return(Rcpp::wrap( ds )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>")
print( res <- funx() )
stopifnot( identical( res, as.numeric(0:1)))
cat("\n=== set<raw>\n")
foo <- '
std::set<Rbyte> bs ;
bs.insert( (Rbyte)0 ) ;
bs.insert( (Rbyte)1 ) ;
bs.insert( (Rbyte)0 ) ;
return(Rcpp::wrap( bs )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>")
print( res <- funx() )
stopifnot( identical( res, as.raw(0:1)))
cat("\n=== set<string> \n")
foo <- '
std::set<std::string> ss ;
```

```
ss.insert( "foo" ) ;
ss.insert( "bar" ) ;
ss.insert( "foo" ) ;
return(Rcpp::wrap( ss )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, include = "#include <set>" )
print( res <- funx() )
stopifnot( identical( res, c("bar","foo")) )
#========= attributes
funx <- cfunction(
        signature(x="data.frame"), '
std::vector<std::string> iv = Rcpp::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 )
```

8.7 RcppInline/RcppSimpleExample.r

A very simple example of using inline without Rcpp.

```
#!/usr/bin/env r
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
foo <- '
  int i, j, na, nb, nab;
  double *xa, *xb, *xab;
  SEXP ab;
  PROTECT(a = AS_NUMERIC(a));
  PROTECT(b = AS_NUMERIC(b));
  na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
  PROTECT(ab = NEW_NUMERIC(nab));
  xa = NUMERIC_POINTER(a); xb = NUMERIC_POINTER(b);
  xab = NUMERIC_POINTER(ab);
  for(i = 0; i < nab; i++) xab[i] = 0.0;
  for(i = 0; i < na; i++)
    for(j = 0; j < nb; j++) xab[i + j] += xa[i] * xb[j];
  UNPROTECT(3);
  return(ab);
'
funx <- cfunction(signature(a="numeric",b="numeric"), foo, Rcpp=FALSE, verbose=FALSE)
funx(a=1:20, b=2:11)
```

8.8 RcppInline/RcppSimpleTests.r

This file provides a few simple tests that preceded the creation of the numerous formal unit tests that followed.

8.9 RcppInline/UncaughtExceptions.r

An example of how to catch C++ exceptions even without a try / catch block.

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010  Romain Francois and Dirk Eddelbuettel
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
require(Rcpp)
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
require(inline)
funx_old <- cxxfunction(
        signature(),
        ’throw std::range_error("boom"); return R_NilValue ; ’,
        plugin = "Rcpp" )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
cppFunction(’
SEXP funx(){
    throw std::range_error("boom"); return R_NilValue ;
}’)
tryCatch(  funx(), "C++Error" = function(e){
        cat( sprintf( "C++ exception of class ’%s’ : %s\n", class(e)[1L], e$message  ) )
} )
# or using a direct handler
tryCatch(  funx(), "std::range_error" = function(e){
        cat( sprintf( "C++ exception of class ’%s’ : %s\n", class(e)[1L], e$message  ) )
} )
# just to check things carry on
print( rnorm(10) )
```

8.10 ConvolveBenchmarks/convolve2_c.c

The basic convolution example from Section 5.10.1 of 'Writing R Extensions'

```
/* This is from ’Writing R Extensions’ section 5.10.1 */
#include <R.h>
#include <Rdefines.h>
SEXP convolve2(SEXP a, SEXP b)
{
    int i, j, na, nb, nab;
    double *xa, *xb, *xab;
    SEXP ab;
    PROTECT(a = AS_NUMERIC(a));
    PROTECT(b = AS_NUMERIC(b));
    na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
    PROTECT(ab = NEW_NUMERIC(nab));
    xa = NUMERIC_POINTER(a); xb = NUMERIC_POINTER(b);
    xab = NUMERIC_POINTER(ab);
    for(i = 0; i < nab; i++) xab[i] = 0.0;
    for(i = 0; i < na; i++)
        for(j = 0; j < nb; j++) xab[i + j] += xa[i] * xb[j];
    UNPROTECT(3);
    return(ab);
}
#include "loopmacro.h"
LOOPMACRO_C(convolve2)
```

8.11 ConvolveBenchmarks/convolve2_cpp.cpp

A simple version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp using RcppVector<double>.

8.12 ConvolveBenchmarks/convolve3_cpp.cpp

A more efficient version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp and using Rcpp::NumericVector.

```
// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
// This is a rewrite of the ’Writing R Extensions’ section 5.10.1 example
#include <Rcpp.h>
RcppExport SEXP convolve3cpp(SEXP a, SEXP b){
    Rcpp::NumericVector xa(a);
    Rcpp::NumericVector xb(b);
    int n_xa = xa.size() ;
    int n_xb = xb.size() ;
    int nab = n_xa + n_xb - 1;
    Rcpp::NumericVector xab(nab);
    for (int i = 0; i < n_xa; i++)
        for (int j = 0; j < n_xb; j++)
            xab[i + j] += xa[i] * xb[j];
    return xab ;
}
#include "loopmacro.h"
LOOPMACRO_CPP(convolve3cpp)
```

8.13 ConvolveBenchmarks/convolve4_cpp.cpp

An even more efficient version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp and using Rcpp::NumericVector as well as direct pointer operations for better performance.

```
// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
// This is a rewrite of the ’Writing R Extensions’ section 5.10.1 example
#include <Rcpp.h>
RcppExport SEXP convolve4cpp(SEXP a, SEXP b) {
    Rcpp::NumericVector xa(a);
    Rcpp::NumericVector xb(b);
    int n_xa = xa.size() ;
    int n_xb = xb.size() ;
    int nab = n_xa + n_xb - 1;
    Rcpp::NumericVector xab(nab,0.0);
    double* pa = xa.begin() ;
    double* pb = xb.begin() ;
    double* pab = xab.begin() ;
    int i,j=0;
    for (i = 0; i < n_xa; i++)
        for (j = 0; j < n_xb; j++)
            pab[i + j] += pa[i] * pb[j];
    return xab ;
}
#include "loopmacro.h"
LOOPMACRO_CPP(convolve4cpp)
```

8.14 ConvolveBenchmarks/convolve7_c.c

The basic convolution example from Section 5.10.1 of 'Writing R Extensions', written using REAL(x)[i] accessor macros to demonstrate the performance hit imposed by these.

```
// This is from ’Writing R Extensions’ section 5.10.1
// BUT slowed down by using REAL() on each access which proves to be rather costly
```

```
#include <R.h>
#include <Rdefines.h>
SEXP convolve7(SEXP a, SEXP b)
{
    int i, j, na, nb, nab;
    SEXP ab;
    PROTECT(a = AS_NUMERIC(a));
    PROTECT(b = AS_NUMERIC(b));
    na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
    PROTECT(ab = NEW_NUMERIC(nab));
    for(i = 0; i < nab; i++) REAL(ab)[i] = 0.0;
    for(i = 0; i < na; i++)
        for(j = 0; j < nb; j++) REAL(ab)[i + j] += REAL(a)[i] * REAL(b)[j];
    UNPROTECT(3);
    return(ab);
}
#include "loopmacro.h"
LOOPMACRO_C(convolve7)
```

8.15 ConvolveBenchmarks/exampleRCode.r

R file / littler script to run and time the various implementations.

```
#!/usr/bin/env r
suppressMessages(require(Rcpp))
set.seed(42)
n <- 200
a <- rnorm(n)
b <- rnorm(n)
## load shared libraries with wrapper code
dyn.load("convolve2_c.so")
dyn.load("convolve3_cpp.so")
dyn.load("convolve4_cpp.so")
dyn.load("convolve5_cpp.so")
dyn.load("convolve7_c.so")
dyn.load("convolve8_cpp.so")
dyn.load("convolve9_cpp.so")
dyn.load("convolve10_cpp.so")
dyn.load("convolve11_cpp.so")
dyn.load("convolve12_cpp.so" )
dyn.load("convolve14_cpp.so" )
## now run each one once for comparison of results,
## and define test functions
R_API_optimised <- function(n,a,b) .Call("convolve2__loop", n, a, b)
Rcpp_New_std <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b)
#Rcpp_New_std_inside <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b, PACKAGE = "Rcpp" )
Rcpp_New_ptr <- function(n,a,b) .Call("convolve4cpp__loop", n, a, b)
Rcpp_New_sugar <- function(n,a,b) .Call("convolve5cpp__loop", n, a, b)
Rcpp_New_sugar_noNA <- function(n,a,b) .Call("convolve11cpp__loop", n, a, b)
R_API_naive <- function(n,a,b) .Call("convolve7__loop", n, a, b)
Rcpp_New_std_2 <- function(n,a,b) .Call("convolve8cpp__loop", n, a, b)
#Rcpp_New_std_3 <- function(n,a,b) .Call("convolve9cpp__loop", n, a, b)
#Rcpp_New_std_4 <- function(n,a,b) .Call("convolve10cpp__loop", n, a, b)
Rcpp_New_std_it <- function(n,a,b) .Call("convolve12cpp__loop", n, a, b )
Rcpp_New_std_Fast <- function(n,a,b) .Call("convolve14cpp__loop", n, a, b )
v1 <- R_API_optimised(1L, a, b )
v3 <- Rcpp_New_std(1L, a, b)
v4 <- Rcpp_New_ptr(1L, a, b)
v5 <- Rcpp_New_sugar(1L, a, b )
v7 <- R_API_naive(1L, a, b)
v11 <- Rcpp_New_sugar_noNA(1L, a, b)
stopifnot(all.equal(v1, v3))
stopifnot(all.equal(v1, v4))
stopifnot(all.equal(v1, v5))
stopifnot(all.equal(v1, v7))
stopifnot(all.equal(v1, v11))
## load benchmarkin helper function
suppressMessages(library(rbenchmark))
REPS <- 5000L
bm <- benchmark(R_API_optimised(REPS,a,b),
                R_API_naive(REPS,a,b),
                Rcpp_New_std(REPS,a,b),
#                Rcpp_New_std_inside(REPS,a,b),
                Rcpp_New_ptr(REPS,a,b),
                Rcpp_New_sugar(REPS,a,b),
                Rcpp_New_sugar_noNA(REPS,a,b),
                Rcpp_New_std_2(REPS,a,b),
#                Rcpp_New_std_3(REPS,a,b),
#                Rcpp_New_std_4(REPS,a,b),
                Rcpp_New_std_it(REPS,a,b),
                Rcpp_New_std_Fast(REPS,a,b),
                columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
                order="relative",
                replications=1)
print(bm)
cat("All results are equal\n") # as we didn't get stopped
q("no")
sizes <- 1:10*100
REPS <- 5000L
timings <- lapply( sizes, function(size){
    cat( "size = ", size, "..." )
    a <- rnorm(size); b <- rnorm(size)
    bm <- benchmark(R_API_optimised(REPS,a,b),
                R_API_naive(REPS,a,b),
                Rcpp_New_std(REPS,a,b),
                Rcpp_New_ptr(REPS,a,b),
                Rcpp_New_sugar(REPS,a,b),
                Rcpp_New_sugar_noNA(REPS,a,b),
                columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
                order="relative",
                replications=1)
     cat( "  done\n" )
     bm
} )
for( i in seq_along(sizes)){
    timings[[i]]$size <- sizes[i]
}
timings <- do.call( rbind, timings )
require( lattice )
png( "elapsed.png", width = 800, height = 600 )
xyplot( elapsed ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
dev.off()
png( "relative.png", width = 800, height = 600 )
xyplot( relative ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
dev.off()
```

8.16 FastLM/benchmark.r

Linear model benchmark master file

```
#!/usr/bin/env r
#
# Comparison benchmark
#
# This shows how Armadillo improves on the previous version using GNU GSL,
# and how both are doing better than lm.fit()
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
suppressMessages(library(RcppGSL))
suppressMessages(library(RcppArmadillo))
source("lmArmadillo.R")
source("lmGSL.R")
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)
```

```
y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
lmgsl <- lmGSL()
lmarma <- lmArmadillo()
tlm <- mean(replicate(N, system.time( lmfit <- lm(y ~ X - 1) )["elapsed"]), trim=0.05)
tlmfit <- mean(replicate(N, system.time(lmfitfit <- lm.fit(X, y))["elapsed"]), trim=0.05)
tlmgsl <- mean(replicate(N, system.time(lmgsl(y, X))["elapsed"]), trim=0.05)
tlmarma <- mean(replicate(N, system.time(lmarma(y, X))["elapsed"]), trim=0.05)
res <- c(tlm, tlmfit, tlmgsl, tlmarma)
data <- data.frame(results=res, ratios=tlm/res)
rownames(data) <- c("lm", "lm.fit", "lmGSL", "lmArma")
cat("For n=", n, " and k=", k, "\n", sep="")
print(t(data))
print(t(1/data[,1,drop=FALSE])) # regressions per second
```

8.17 FastLM/fastLMviaArmadillo.r

Runs lm via Armadillo and times the run

```
#!/usr/bin/env r
#
# A faster lm() replacement based on Armadillo
#
# This improves on the previous version using GNU GSL
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
source("lmArmadillo.R")
checkLmArmadillo <- function(y, X) {
    fun <- lmArmadillo()
    res <- fun(y, X)
    fit <- lm(y ~ X - 1)
    rc <- all.equal( as.numeric(res[[1]]), as.numeric(coef(fit))) &
          all.equal( as.numeric(res[[2]]), as.numeric(coef(summary(fit))[,2]))
    invisible(rc)
}
timeLmArmadillo <- function(y, X, N) {
    fun <- lmArmadillo();
    meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
}
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)
y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
stopifnot(checkLmArmadillo(y, X))
mt <- timeLmArmadillo(y, X, N)
cat("Armadillo: Running", N, "simulations yields (trimmed) mean time", mt, "\n")
```

8.18 FastLM/fastLMviaGSL.r

Runs lm via GSL and times the run

```
#!/usr/bin/env r
#
```

```
# A faster lm() replacement based on GNU GSL
#
# This first appeared in the 'Intro to HPC tutorials'
# but has been wrapped in inline::cfunction() here
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
source("lmGSL.R")
checkLmGSL <- function(y, X) {
    fun <- lmGSL()
    res <- fun(y, X)
    fit <- lm(y ~ X - 1)
    rc <- all.equal( res[[1]], as.numeric(coef(fit))) &
          all.equal( res[[2]], as.numeric(coef(summary(fit))[,2]))
    invisible(rc)
}
timeLmGSL <- function(y, X, N) {
    fun <- lmGSL();
    meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
}
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)
y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
stopifnot(checkLmGSL(y, X))
mt <- timeLmGSL(y, X, N)
cat("GSL: Running", N, "simulations yields (trimmed) mean time", mt, "\n")
```

8.19 FastLM/lmArmadillo.r

Armadillo-based implementation of a bare-boned lm()

8.20 FastLM/lmGSL.r

GSL-based implementation of a bare-boned lm()

8.21 SugarPerformance/sugarBenchmarks.R

Benchmarking the 'Rcpp sugar' extensions

```
#!/usr/bin/env r
suppressMessages(library(inline))
suppressMessages(library(Rcpp))
benchmark <- function(start = settings$start,
                      hand.written = settings$hand.written,
                      sugar = settings$sugar,
                      expr = settings$expr,
                      runs = settings$runs,
                      data = settings$data,
                      end = settings$end,
                      inc = settings$inc,

                      settings = list(
                                start = "", hand.written = "",
                                sugar = "", expr = NULL,
                                runs = 500,
                                data = NULL ,
                                end = "",
                                inc = ""
                                )
                      ) {
expr <- force(expr)
inc  <- force( inc )
src <- sprintf( ’
    unsigned int runs = as<int>(runss);
        Environment e(env) ;
        %s
    Timer timer;
    // approach one
    timer.Start();
    for (unsigned int i=0; i<runs; i++) {
           %s
    }
    timer.Stop();
    double t1 = timer.ElapsedTime();
    // approach two
    timer.Reset(); timer.Start();
    for (unsigned int i=0; i<runs; i++) {
        %s
    }
    timer.Stop();
    double t2 = timer.ElapsedTime();
    Language call(expr) ;
    timer.Reset(); timer.Start();
    for (unsigned int i=0; i<runs; i++) {
        NumericVector res2 = Rcpp_fast_eval( call, e ) ;
    }
    timer.Stop();
    double t3 = timer.ElapsedTime();
    %s
    return NumericVector::create(
        _["hand written"] = t1,
        _["sugar"] = t2,
        _["R"]     = t3
        ) ;
’,
               paste( start, collapse = "\n" ) ,
               paste( hand.written, collapse = "\n" ),
               paste( sugar, collapse = "\n" ),
               paste( end, collapse = "\n" )
        )
        e <- environment()
        for( i in names(data) ){
                assign( i, data[[i]], envir = e )
        }
        settings <- getPlugin("Rcpp")
        settings$env$PKG_CXXFLAGS <- paste("-I", getwd(), sep="")
        fun <- cxxfunction(signature(runss="integer", expr = "language", env = "environment" ),
                           src,
                           includes= sprintf( ’#include "Timer.h"\n%s’, paste( inc, collapse = "\n" ) ),
                           plugin="Rcpp",
                           settings=settings)
        results <- fun(runs, expr, environment() )
        cat( "-" )
        list( results = results, runs = runs, expr = deparse(expr) )
}
settings.ifelse <- list( start = ’
        NumericVector x = e["x"] ;
        NumericVector y = e["y"] ;
’, hand.written = ’
        int n = x.size() ;
        NumericVector res1( n ) ;
        double x_ = 0.0 ;
        double y_ = 0.0 ;
        for( int i=0; i<n; i++){
        x_ = x[i] ;
        y_ = y[i] ;
        if( R_IsNA(x_) || R_IsNA(y_) ){
            res1[i] = NA_REAL;
        } else if( x_ < y_ ){
            res1[i] = x_ * x_ ;
        } else {
            res1[i] = -( y_ * y_)  ;
        }
    }
', sugar = '
    NumericVector res2 = ifelse( x < y, x*x, -(y*y) ) ;
', expr = quote(ifelse(x<y, x*x, -(y*y) )),
   data = list( x = runif(1e5),  y = runif(1e5) )
)
settings.ifelse.nona <- list( start = '
        NumericVector x = e["x"] ;
        NumericVector y = e["y"] ;
', hand.written = '
        int n = x.size() ;
        NumericVector res1( n ) ;
        double x_ = 0.0 ;
        double y_ = 0.0 ;
        for( int i=0; i<n; i++){
        x_ = x[i] ;
        y_ = y[i] ;
        if( x_ < y_ ){
            res1[i] = x_ * x_ ;
        } else {
            res1[i] = -( y_ * y_)  ;
        }
    }
', sugar = '
    NumericVector res2 = ifelse( x < y, noNA(x)*noNA(x), -(noNA(y)*noNA(y)) ) ;
', expr = quote(ifelse(x<y, x*x, -(y*y) )),
   data = list( x = runif(1e5),  y = runif(1e5) )
)
settings.sapply <- list( start =  '
        NumericVector x = e["x"] ;
        int n = x.size() ;
', hand.written = '
        NumericVector res1( n ) ;
        std::transform( x.begin(), x.end(), res1.begin(), square ) ;
', sugar = '
        NumericVector res2 = sapply( x, square ) ;
',
        expr = quote(sapply(x,square)),
        runs = 500,
        data = list(
                x = rnorm(1e5) ,
                square = function(x) x*x
        ),
        inc = '
        inline double square(double x){ return x*x ; }
        '
)
settings.any <- list( start = '
        NumericVector x = e["x"] ;
        NumericVector y = e["y"] ;
        int res ;
        SEXP res2 ;
', hand.written = '
        int n = x.size() ;
        bool seen_na = false ;
        bool result = false ;
        double x_ = 0.0 ;
        double y_ = 0.0 ;
        for( int i=0; i<n; i++){
                x_ = x[i] ;
                if( R_IsNA( x_ )  ){
                        seen_na = true ;
                } else {
                        y_ = y[i] ;
                        if( R_IsNA( y_ ) ){
                                seen_na = true ;
                        } else {
                                /* both non NA */
                                if( x_*y_ < 0.0 ){
                                        result = true ;
                                        break ;
                                }
                        }
                }
        }
        res = result ? TRUE : ( seen_na ? NA_LOGICAL : FALSE ) ;
', sugar = '
        res2 = any( x*y < 0 ) ;
',
        expr = quote(any(x*y<0)),
        runs = 5000,
        data = list(
                x = seq( -1, 1, length = 1e05),
                y = rep( 1, 1e05)
        )
)
raw.results <- list(
        benchmark( settings = settings.any   , runs = 5000 ),
        benchmark( settings = settings.ifelse, runs = 500 ),
        benchmark( settings = settings.ifelse.nona, runs = 500 ),
        benchmark( settings = settings.sapply, runs = 500 )
)
cat("\n")
results <- do.call( rbind, lapply( raw.results, "[[", "results" ) )
results <- data.frame(
        runs = sapply( raw.results, "[[", "runs" ),
        expr = sapply( raw.results, "[[", "expr" ),
        as.data.frame( results, stringsAsFactors = FALSE )
        )
results[[ "hand/sugar" ]] <- results[["hand.written" ]] / results[["sugar"]]
results[[ "R/sugar" ]]    <- results[["R" ]]            / results[["sugar"]]
# results <- results[ order( results[["expr"]], results[["runs"]] ), ]
options( width = 300 )
print( results )
```

8.22 SugarPerformance/Timer.h

Simple C++ timer class in a headers-only implementation

```
// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
//
// Timer.h: Rcpp R/C++ interface class library -- simple timer class
//
// Copyright (C) 2010   Dirk Eddelbuettel and Romain Francois
//
// This file is part of Rcpp.
//
// Rcpp is free software: you can redistribute it and/or modify it
// under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 2 of the License, or
// (at your option) any later version.
//
// Rcpp is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Rcpp.  If not, see <http://www.gnu.org/licenses/>.
// Simple timer class based on 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
```

8.23 SugarPerformance/Timertest.cpp

Simple test program for the C++ timer class

```
// -*- mode: c++; compile-command: "g++ -Wall -O3 -o Timertest Timertest.cpp"; c-indent-level: 4;
       c-basic-offset: 4; tab-width: 8 -*-
// from http://www.cs.uiowa.edu/~sriram/30/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;
}
```

8.24 OpenMP/piWithInterrupts.cpp.R

Computing pi via MPI, and letting the user interupt via Ctrl-C

8.25 OpenMP/check.R

Compares both approaches

```
dyn.load("piWithInterrupts.so")
#res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
print(res, digits=10)
```

8.26 OpenMP/OpenMPandInline.r

Example of using OpenMP compilation via inline

```
#!/usr/bin/env r
library(inline)
library(rbenchmark)
serialCode <- '
   // assign to C++ vector
   std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
   size_t n = x.size();
   for (size_t i=0; i<n; i++) {
       x[i] = ::log(x[i]);
   }
   return Rcpp::wrap(x);
’
funSerial <- cxxfunction(signature(xs="numeric"), body=serialCode, plugin="Rcpp")
serialStdAlgCode <- ’
   std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
   std::transform(x.begin(), x.end(), x.begin(), ::log);
   return Rcpp::wrap(x);
’
funSerialStdAlg <- cxxfunction(signature(xs="numeric"), body=serialStdAlgCode, plugin="Rcpp")
## same, but with Rcpp vector just to see if there is measurable difference
serialRcppCode <- ’
   // assign to C++ vector
   Rcpp::NumericVector x = Rcpp::NumericVector(xs);
   size_t n = x.size();
   for (size_t i=0; i<n; i++) {
       x[i] = ::log(x[i]);
   }
   return x;
’
funSerialRcpp <- cxxfunction(signature(xs="numeric"), body=serialRcppCode, plugin="Rcpp")
serialStdAlgRcppCode <- ’
   Rcpp::NumericVector x = Rcpp::NumericVector(xs);
   std::transform(x.begin(), x.end(), x.begin(), ::log);
   return x;
’
funSerialStdAlgRcpp <- cxxfunction(signature(xs="numeric"), body=serialStdAlgRcppCode, plugin="Rcpp")
serialImportTransRcppCode <- ’
   Rcpp::NumericVector x(xs);
   return Rcpp::NumericVector::import_transform(x.begin(), x.end(), ::log);
’
funSerialImportTransRcpp <- cxxfunction(signature(xs="numeric"), body=serialImportTransRcppCode, plugin="Rcpp")
## now with a sugar expression with internalizes the loop
sugarRcppCode <- ’
   // assign to C++ vector
   Rcpp::NumericVector x = log ( Rcpp::NumericVector(xs) );
   return x;
’
funSugarRcpp <- cxxfunction(signature(xs="numeric"), body=sugarRcppCode, plugin="Rcpp")
## lastly via OpenMP for parallel use
openMPCode <- ’
   // assign to C++ vector
   std::vector<double> x = Rcpp::as<std::vector< double > >(xs);
   size_t n = x.size();
#pragma omp parallel for shared(x, n)
   for (size_t i=0; i<n; i++) {
       x[i] = ::log(x[i]);
   }
   return Rcpp::wrap(x);
’
## modify the plugin for Rcpp to support OpenMP
settings <- getPlugin("Rcpp")
settings$env$PKG_CXXFLAGS <- paste(’-fopenmp’, settings$env$PKG_CXXFLAGS)
settings$env$PKG_LIBS <- paste(’-fopenmp -lgomp’, settings$env$PKG_LIBS)
funOpenMP <- cxxfunction(signature(xs="numeric"), body=openMPCode, plugin="Rcpp", settings=settings)
z <- seq(1, 2e6)
res <- benchmark(funSerial(z), funSerialStdAlg(z),
                 funSerialRcpp(z), funSerialStdAlgRcpp(z),
                 funSerialImportTransRcpp(z),
                 funOpenMP(z), funSugarRcpp(z),
                 columns=c("test", "replications", "elapsed",
                           "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=100)
print(res)
```

8.27 RcppGibbs/RcppGibbs.R

Implementation of a MCMC Gibbs sampler using Rcpp

```
## Simple Gibbs Sampler Example
## Adapted from Darren Wilkinson’s post at
## http://darrenjw.wordpress.com/2010/04/28/mcmc-programming-in-r-python-java-and-c/
##
## Sanjog Misra and Dirk Eddelbuettel, June-July 2011
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
suppressMessages(library(compiler))
suppressMessages(library(rbenchmark))
## Actual joint density -- the code which follow implements
## a Gibbs sampler to draw from the following joint density f(x,y)
fun <- function(x,y) {
    x*x * exp(-x*y*y - y*y + 2*y - 4*x)
}
## Note that the full conditionals are propotional to
## f(x|y) = (x^2)*exp(-x*(4+y*y))              : a Gamma density kernel
## f(y|x) = exp(-0.5*2*(x+1)*(y^2 - 2*y/(x+1)) : Normal Kernel
## There is a small typo in Darrens code.
## The full conditional for the normal has the wrong variance
## It should be 1/sqrt(2*(x+1)) not 1/sqrt(1+x)
## This we can verify ...
## The actual conditional (say for x=3) can be computed as follows
## First - Construct the Unnormalized Conditional
fy.unnorm <- function(y) fun(3,y)
## Then - Find the appropriate Normalizing Constant
K <- integrate(fy.unnorm,-Inf,Inf)
## Finally - Construct Actual Conditional
fy <- function(y) fy.unnorm(y)/K$val
## Now - The corresponding Normal should be
fy.dnorm <- function(y) {
    x <- 3
    dnorm(y,1/(1+x),sqrt(1/(2*(1+x))))
}
## and not ...
fy.dnorm.wrong <- function(y) {
    x <- 3
    dnorm(y,1/(1+x),sqrt(1/((1+x))))
}
if (interactive()) {
    ## Graphical check
    ## Actual (gray thick line)
    curve(fy,-2,2,col='grey',lwd=5)
    ## Correct Normal conditional (blue dotted line)
    curve(fy.dnorm,-2,2,col='blue',add=T,lty=3)
    ## Wrong Normal (Red line)
    curve(fy.dnorm.wrong,-2,2,col='red',add=T)
}
## Here is the actual Gibbs Sampler
## This is Darren Wilkinsons R code (with the corrected variance)
## But we are returning only his columns 2 and 3 as the 1:N sequence
## is never used below
Rgibbs <- function(N,thin) {
    mat <- matrix(0,ncol=2,nrow=N)
    x <- 0
    y <- 0
    for (i in 1:N) {
        for (j in 1:thin) {
            x <- rgamma(1,3,y*y+4)
            y <- rnorm(1,1/(x+1),1/sqrt(2*(x+1)))
        }
        mat[i,] <- c(x,y)
    }
    mat
}
## We can also try the R compiler on this R function
RCgibbs <- cmpfun(Rgibbs)
## For example
## mat <- Rgibbs(10000,10); dim(mat)
## would give: [1] 10000     2
## Now for the Rcpp version -- Notice how easy it is to code up!
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
gibbscode <- '
  using namespace Rcpp;   // inline does that for us already
  // n and thin are SEXPs which the Rcpp::as function maps to C++ vars
  int N   = as<int>(n);
  int thn = as<int>(thin);
  int i,j;
  NumericMatrix mat(N, 2);
  RNGScope scope;         // Initialize Random number generator. Not needed when Attributes used.
  // The rest of the code follows the R version
  double x=0, y=0;
  for (i=0; i<N; i++) {
    for (j=0; j<thn; j++) {
      x = ::Rf_rgamma(3.0,1.0/(y*y+4));
      y = ::Rf_rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
    }
    mat(i,0) = x;
    mat(i,1) = y;
  }
  return mat;             // Return to R
’
# Compile and Load
RcppGibbs_old <- cxxfunction(signature(n="int", thin = "int"),
                         gibbscode, plugin="Rcpp")
gslgibbsincl <- ’
  #include <gsl/gsl_rng.h>
  #include <gsl/gsl_randist.h>
  using namespace Rcpp;  // just to be explicit
’
gslgibbscode <- ’
  int N = as<int>(ns);
  int thin = as<int>(thns);
  int i, j;
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  double x=0, y=0;
  NumericMatrix mat(N, 2);
  for (i=0; i<N; i++) {
    for (j=0; j<thin; j++) {
      x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
      y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
    }
    mat(i,0) = x;
    mat(i,1) = y;
  }
  gsl_rng_free(r);
  return mat;           // Return to R
’
## Compile and Load
GSLGibbs_old <- cxxfunction(signature(ns="int", thns = "int"),
                        body=gslgibbscode, includes=gslgibbsincl,
                        plugin="RcppGSL")
## without RcppGSL, using cfunction()
#GSLGibbs <- cfunction(signature(ns="int", thns = "int"),
#                      body=gslgibbscode, includes=gslgibbsincl,
#                      Rcpp=TRUE,
#                      cppargs="-I/usr/include",
#                      libargs="-lgsl -lgslcblas")
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
## Use of the cppFunction() gives the ability to immediately compile embed C++
## without having to worry about header specification or Rcpp attributes.
cppFunction(’
NumericMatrix RcppGibbs(int N, int thn){
    // Note: n and thin are SEXPs which the Rcpp automatically converts to ints
    // Setup storage matrix
    NumericMatrix mat(N, 2);
    // The rest of the code follows the R version
    double x = 0, y = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < thn; j++) {
            x = R::rgamma(3.0,1.0/(y*y+4));
            y = R::rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
        }
        mat(i,0) = x;
        mat(i,1) = y;
    }
    return mat;             // Return to R
}’)
## Use of the sourceCpp() is preferred for users who wish to source external
## files or specify their headers and Rcpp attributes within their code.
## Code here is able to easily be extracted and placed into its own C++ file.
## Compile and Load
sourceCpp(code="
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp;  // just to be explicit
// [[Rcpp::depends(RcppGSL)]]
// [[Rcpp::export]]
NumericMatrix GSLGibbs(int N, int thin){
    gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
    double x = 0, y = 0;
    NumericMatrix mat(N, 2);
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < thin; j++) {
            x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
            y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
        }
        mat(i,0) = x;
        mat(i,1) = y;
    }
    gsl_rng_free(r);
    return mat;           // Return to R
}")
## Now for some tests
## You can try other values if you like
## Note that the total number of interations are N*thin!
Ns <- c(1000,5000,10000,20000)
thins <- c(10,50,100,200)
tim_R <- rep(0,4)
tim_RC <- rep(0,4)
tim_Rgsl <- rep(0,4)
tim_Rcpp <- rep(0,4)
for (i in seq_along(Ns)) {
    tim_R[i] <- system.time(mat <- Rgibbs(Ns[i],thins[i]))[3]
    tim_RC[i] <- system.time(cmat <- RCgibbs(Ns[i],thins[i]))[3]
    tim_Rgsl[i] <- system.time(gslmat <- GSLGibbs(Ns[i],thins[i]))[3]
    tim_Rcpp[i] <- system.time(rcppmat <- RcppGibbs(Ns[i],thins[i]))[3]
    cat("Replication #", i, "complete \n")
}
## Comparison
speedup <- round(tim_R/tim_Rcpp,2);
speedup2 <- round(tim_R/tim_Rgsl,2);
speedup3 <- round(tim_R/tim_RC,2);
summtab <- round(rbind(tim_R,tim_RC, tim_Rcpp,tim_Rgsl,speedup3,speedup,speedup2),3)
colnames(summtab) <- c("N=1000","N=5000","N=10000","N=20000")
rownames(summtab) <- c("Elasped Time (R)","Elasped Time (RC)","Elapsed Time (Rcpp)", "Elapsed Time (Rgsl)",
                       "SpeedUp Rcomp.","SpeedUp Rcpp", "SpeedUp GSL")
print(summtab)
## Contour Plots -- based on Darren’s example
if (interactive() && require(KernSmooth)) {
    op <- par(mfrow=c(4,1),mar=c(3,3,3,1))
    x <- seq(0,4,0.01)
    y <- seq(-2,4,0.01)
    z <- outer(x,y,fun)
    contour(x,y,z,main="Contours of actual distribution",xlim=c(0,2), ylim=c(-2,4))
    fit <- bkde2D(as.matrix(mat),c(0.1,0.1))
    contour(drawlabels=T, fit$x1, fit$x2, fit$fhat, xlim=c(0,2), ylim=c(-2,4),
            main=paste("Contours of empirical distribution:",round(tim_R[4],2)," seconds"))
    fitc <- bkde2D(as.matrix(rcppmat),c(0.1,0.1))
    contour(fitc$x1,fitc$x2,fitc$fhat,xlim=c(0,2), ylim=c(-2,4),
            main=paste("Contours of Rcpp based empirical distribution:",round(tim_Rcpp[4],2)," seconds"))
    fitg <- bkde2D(as.matrix(gslmat),c(0.1,0.1))
    contour(fitg$x1,fitg$x2,fitg$fhat,xlim=c(0,2), ylim=c(-2,4),
            main=paste("Contours of GSL based empirical distribution:",round(tim_Rgsl[4],2)," seconds"))
    par(op)
}
## also use rbenchmark package
N <- 20000
thn <- 200
res <- benchmark(Rgibbs(N, thn),
                 RCgibbs(N, thn),
                 RcppGibbs(N, thn),
                 GSLGibbs(N, thn),
                 columns=c("test", "replications", "elapsed",
                           "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=10)
print(res)
## And we are done
```

8.28 RcppGibbs/timeRNGs.R

Illustration of speed differences between Nornal and Uniform RNG draws from R and GSL

```
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
suppressMessages(library(rbenchmark))
```

```
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project, and
## do NOT use the old code below
cppFunction(’
NumericVector rcppGamma(NumericVector x){
    int n   = x.size();
    const double y = 1.234;
    for (int i=0; i<n; i++) {
        x[i] = R::rgamma(3.0, 1.0/(y*y+4));
    }
    // Return to R
    return x;
}’)
## This approach is a bit sloppy. Generally, you will want to use
## sourceCpp() if there are additional includes that are required.
cppFunction(’
NumericVector gslGamma(NumericVector x){
    int n   = x.size();
    gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
    const double y = 1.234;
    for (int i=0; i<n; i++) {
        x[i] = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
    }
    gsl_rng_free(r);
    // Return to R
    return x;
}’, includes = ’#include <gsl/gsl_rng.h>
                 #include <gsl/gsl_randist.h>’,
    depends = "RcppGSL")
cppFunction(’
NumericVector rcppNormal(NumericVector x){
    int n   = x.size();
    const double y = 1.234;
    for (int i=0; i<n; i++) {
        x[i] = R::rnorm(1.0/(y+1),1.0/sqrt(2*y+2));
    }
    // Return to R
    return x;
}’)
## Here we demonstrate the use of sourceCpp() to show the continuity
## of the code artifact.
sourceCpp(code = ’
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp;
// [[Rcpp::depends("RcppGSL")]]
// [[Rcpp::export]]
NumericVector gslNormal(NumericVector x){
    int n   = x.size();
    gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
    const double y = 1.234;
    for (int i=0; i<n; i++) {
        x[i] = 1.0/(y+1)+gsl_ran_gaussian(r,1.0/sqrt(2*y+2));
    }
    gsl_rng_free(r);
    // Return to R
    return x;
}’)
x <- rep(NA, 1e6)
res <- benchmark(rcppGamma(x),
                  gslGamma(x),
                  rcppNormal(x),
                  gslNormal(x),
                  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                  order="relative",
                  replications=20)
print(res)
##
##
##  Old code below. Do not use in new projects, it is here solely for comparison
##
##
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
rcppGamma_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body=’
  NumericVector x(xs);
  int n   = x.size();
  RNGScope scope;         // Initialize Random number generator. Not needed when Attributes used.
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = ::Rf_rgamma(3.0, 1.0/(y*y+4));
  }
  // Return to R
  return x;
')
gslGamma_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
                        include='#include <gsl/gsl_rng.h>
                                 #include <gsl/gsl_randist.h>',
                        body='
  NumericVector x(xs);
  int n   = x.size();
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
  }
  gsl_rng_free(r);
  // Return to R
  return x;
')
rcppNormal_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body='
  NumericVector x(xs);
  int n   = x.size();
  RNGScope scope;          // Initialize Random number generator. Not needed when Attributes used.
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = ::Rf_rnorm(1.0/(y+1),1.0/sqrt(2*y+2));
  }
  // Return to R
  return x;
')
gslNormal_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
                        include='#include <gsl/gsl_rng.h>
                                 #include <gsl/gsl_randist.h>',
                        body='
  NumericVector x(xs);
  int n   = x.size();
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = 1.0/(y+1)+gsl_ran_gaussian(r,1.0/sqrt(2*y+2));
  }
  gsl_rng_free(r);
  // Return to R
  return x;
')
```

8.29 Misc/fibonacci.r

Faster recursive computation of Fibonacci numbers via Rcpp

```
#!/usr/bin/env r
## this short example was provided in response to this StackOverflow questions:
## http://stackoverflow.com/questions/6807068/why-is-my-recursive-function-so-slow-in-r
## and illustrates that recursive function calls are a) really expensive in R and b) not
## all expensive in C++ (my machine sees a 700-fold speed increase) and c) the byte
## compiler in R does not help here.
suppressMessages(library(Rcpp))
## byte compiler
require(compiler)
## A C++ version compile with cppFunction
fibRcpp <- cppFunction( '
int fibonacci(const int x) {
   if (x == 0) return(0);
   if (x == 1) return(1);
   return (fibonacci(x - 1)) + fibonacci(x - 2);
}
' )
## for comparison, the original (but repaired with 0/1 offsets)
fibR <- function(seq) {
    if (seq == 0) return(0);
    if (seq == 1) return(1);
    return (fibR(seq - 1) + fibR(seq - 2));
}
## also use byte-compiled R function
```

```
fibRC <- cmpfun(fibR)
## load rbenchmark to compare
library(rbenchmark)
N <- 35     ## same parameter as original post
res <- benchmark(fibR(N),
                 fibRC(N),
                 fibRcpp(N),
                 columns=c("test", "replications", "elapsed",
                           "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=1)
print(res)  ## show result
```

8.30 Misc/ifelseLooped.r

Nice example of accelerating path-dependent loops with Rcpp

```
#!/usr/bin/env r
##
## This example goes back to the following StackOverflow questions:
##   http://stackoverflow.com/questions/7153586/can-i-vectorize-a-calculation-which-depends-on-previous-elements
## and provides a nice example of how to accelerate path-dependent
## loops which are harder to vectorise.  It lead to the following blog
## post:
##   http://dirk.eddelbuettel.com/blog/2011/08/23#rcpp_for_path_dependent_loops
##
## Thanks to Josh Ulrich for provided a first nice (R-based) answer on
## StackOverflow and for also catching a small oversight in my posted answer.
##
## Dirk Eddelbuettel, 23 Aug 2011
##
## Copyrighted but of course GPL'ed
library(inline)
library(rbenchmark)
library(compiler)
fun1 <- function(z) {
  for(i in 2:NROW(z)) {
    z[i] <- ifelse(z[i-1]==1, 1, 0)
  }
  z
}
fun1c <- cmpfun(fun1)
fun2 <- function(z) {
  for(i in 2:NROW(z)) {
    z[i] <- if(z[i-1]==1) 1 else 0
  }
  z
}
fun2c <- cmpfun(fun2)
funRcpp <- cxxfunction(signature(zs="numeric"), plugin="Rcpp", body="
  Rcpp::NumericVector z = Rcpp::NumericVector(zs);
  int n = z.size();
  for (int i=1; i<n; i++) {
    z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
  }
  return(z);
")
z <- rep(c(1,1,0,0,0,0), 100)
## test all others against fun1 and make sure all are identical
all(sapply(list(fun2(z),fun1c(z),fun2c(z),funRcpp(z)), identical, fun1(z)))
res <- benchmark(fun1(z), fun2(z),
                 fun1c(z), fun2c(z),
                 funRcpp(z),
                 columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=1000)
print(res)
z <- c(1,1,0,0,0,0)
res2 <- benchmark(fun1(z), fun2(z),
                 fun1c(z), fun2c(z),
                 funRcpp(z),
                 columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=10000)
print(res2)
if (FALSE) {                            # quick test to see if Int vectors are faster: appears not
```

```
    funRcppI <- cxxfunction(signature(zs="integer"), plugin="Rcpp", body="
        Rcpp::IntegerVector z = Rcpp::IntegerVector(zs);
        int n = z.size();
        for (int i=1; i<n; i++) {
            z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
        }
        return(z);
    ")
    z <- rep(c(1L,1L,0L,0L,0L,0L), 100)
    identical(fun1(z),fun2(z),fun1c(z),fun2c(z),funRcppI(z))
    res3 <- benchmark(fun1(z), fun2(z),
                      fun1c(z), fun2c(z),
                      funRcppI(z),
                      columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                      order="relative",
                      replications=1000)
    print(res3)
    z <- c(1L,1L,0L,0L,0L,0L)
    res4 <- benchmark(fun1(z), fun2(z),
                      fun1c(z), fun2c(z),
                      funRcppI(z),
                      columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                      order="relative",
                      replications=10000)
    print(res4)
}
```