

Package ‘RcppBDT’

August 29, 2016

Type Package

Title Rcpp bindings for the Boost Date_Time library

Version 0.2.3

Date 2014-04-13

Author Dirk Eddelbuettel and Romain Francois

Maintainer Dirk Eddelbuettel <edd@debian.org>

Description This package provides R with access to Boost Date_Time functionality by using Rcpp modules.
Functionality from Boost Date_Time for dates, durations (both for days and datetimes), timezones, and posix time (“ptime”) is provided. The posix time implementation can support high-resolution of up to nano-second precision by using 96 bits (instead of R’s 64) to present a ptime object.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 3.1.0)

Imports Rcpp (>= 0.11.0), methods

LinkingTo Rcpp, BH

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-04-13 23:06:05

R topics documented:

RcppBDT-package	2
bdtDd	2
bdtDt	3
bdtDu	4
bdtPt	5
bdtTz	5
RcppBDT Date functions	6
RcppBDT-constants	7

Index	9
--------------	----------

RcppBDT-package

Bindings for Boost Date_Time

Description

This package provides R with access to Boost Date_Time functionality by using Rcpp modules. Date, Local time, duration and time zone functionality is covered.

Details

Please consult the Boost documentation for (copious) details on the Date_Time library.

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

bdtDd

Rcpp module bdtDd for binding of Boost Date_Time date duration functionality

Description

The bdtDd module is created using Rcpp modules and wraps a helper class bdtDd around Boost Date_time date duration functionality provided by the Boost class `boost::gregorian::date_duration`. New instances can be created using an integer for days of duration.

Usage

```
days(...)  
weeks(...)
```

Arguments

... suitable argument, often an integer, denoting one unit of the reference duration component

Details

Please consult the Boost documentation for (copious) details on the Date_Time library. See the Rcpp-modules vignette for details on Rcpp modules.

Method

show signature(x = "Rcpp_bdtDd"): prints a (BDTdd) date duration class object

format signature(x = "Rcpp_bdtDd"): formats a (BDTdd) date duration class object

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

bdtDt

Rcpp module bdtDt for binding of Boost Date_Time Date functionality

Description

The bdtDt module is created using Rcpp modules and wraps a helper class bdtDt around Boost Date_time date functionality provided by the Boost class boost::gregorian::date.

New instances can be created using either the default constructor (without arguments) or the constructor using year, month, date arguments.

The bdt variable is a default instance of this bdtDt reference class. It facilities accessing the member functions via utility function, see for example [getEndOfBizWeek](#) or `print(bdtDt)` for the available methods.

Details

Please consult the Boost documentation for (copious) details on the Date_Time library. See the Rcpp-modules vignette for details on Rcpp modules.

Method

show signature(x = "Rcpp_bdtDt"): prints a (bdtDt) date class object

format signature(x = "Rcpp_bdtDt"): formats a (bdtDt) date class object

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

bdtDu

Rcpp module bdtDu for binding of Boost Date_Time duration functionality

Description

The bdtDu module is created using Rcpp modules and wraps a helper class bdtDu around Boost Date_time duration functionality provided by the Boost class `boost::posix_time::duration`.

New instances can be created using four integer values for hour, minute, seconds and fractional seconds. Fractional seconds ought to be at a nano-second granularity; there may be platforms not permitting this.

Usage

```
hours(...)
microseconds(...)
milliseconds(...)
minutes(...)
nanoseconds(...)
seconds(...)
```

Arguments

... suitable argument, often an integer, denoting one unit of the reference duration component

Details

Please consult the Boost documentation for (copious) details on the Date_Time library. See the Rcpp-modules vignette for details on Rcpp modules.

Method

show signature(x = "Rcpp_bdtDu"): prints a (BDTdu) duration class object
format signature(x = "Rcpp_bdtDu"): formats a (BDTdu) duration class object

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

bdtPt*Rcpp module bdtPt for binding of Boost Date_Time ptime functionality*

Description

The bdtDu module is created using Rcpp modules and wraps a helper class bdtPt around Boost Date_time duration functionality provided by the Boost class `boost::posix_time::ptime`.

New instances can be created using either a default construction (creating an unset instance) or using seven integer values for year, month, day, hour, minute, seconds and fractional seconds. Fractional seconds ought to be at a nano-second granularity; there may be platforms not permitting this.

Details

Please consult the Boost documentation for (copious) details on the Date_Time library. See the Rcpp-modules vignette for details on Rcpp modules.

Method

show signature(x = "Rcpp_bdtPt"): prints a (bdtPt) ptime class object

format signature(x = "Rcpp_bdtPt"): formats a (bdtPt) ptime class object

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

bdtTz*Rcpp module bdtTz for binding of Boost Date_Time timezone functionality*

Description

The bdtTz module is created using Rcpp modules and wraps a helper class bdtTz around Boost Date_time timezone functionality provided mainly by the Boost classes `boost::local_time::tz_database` and `boost::local_time::time_zone_ptr`.

On startup, the database object is initialized using a local copy (in csv format) of the timezone data. Instances of the timezone object, represented by an instance of the timezone pointer class, can be created and queried.

New instances can be created using a valid timezone region string (such "Europe/London").

Details

Please consult the Boost documentation for (copious) details on the Date_Time library. See the Rcpp-modules vignette for details on Rcpp modules.

Method

show signature(x = "Rcpp_bdtTz"): prints a (bdtTz) timezone class object

format signature(x = "Rcpp_bdtTz"): formats a (bdtTz) timezone class object

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

RcppBDT Date functions

Date accessor and construction functions from Boost Date_Time

Description

This constants are provided for convenience. In the C++ sources, enumeration types are used for days of the week, months of the year as well as the ordering terms.

Similar package-level constants are provided here as well. This should be considered as experimental and may be withdrawn in a later version of the package.

Usage

```
getEndOfBizWeek(date)
getEndOfMonth(date)
getYear(date)
getMonth(date)
getDay(date)
getDayOfWeek(date)
getDayOfYear(date)
getIMMDate(mon, year)
getNthDayOfWeek(nthday, dow, mon, year)
getLastDayOfWeekInMonth(nthday, mon, year)
getFirstDayOfWeekInMonth(nthday, mon, year)
getFirstDayOfWeekAfter(dow, date)
getLastDayOfWeekBefore(dow, date)
```

Arguments

date	a Date object
mon	a month, specified either as an integer or one of the constants Jan , Feb , ... defined in this package
year	a four-digit year, specified as an integer
nthday	either an integer between 1 and 5, or one of the constants first , second , ... fifth defined in this package.
dow	either an integer between 0 and 6 denoting a day of the week, or one of the constants Sun , Mon , ... Sat defined in this package.

Details

Details of the Boost functions are provided by the Boost documentation.

Value

All functions return a [Date](#) object.

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

RcppBDT-constants

Constants for date functions with Boost Date_Time

Description

This constants are provided for convenience. In the C++ sources, enumeration types are used for days of the week, months of the year as well as the ordering terms.

Similar package-level constants are provided here as well. This should be considered as experimental and may be withdrawn in a later version of the package.

Details

Sun, Mon, Tue, ..., Sat can be used instead of the values 0 to 6.

Jan, Feb, ..., Dec can be used instead of the values 1 to 12.

first, second, ..., fifth can be used instead of the values 1 to 5.

We use the same values as the Boost source code. In other words, Sunday is 0, Monday is 1 and so on. Months, however, start at 1 for January.

Author(s)

Dirk Eddelbuettel <edd@debian.org>

References

Boost Date_Time: http://www.boost.org/doc/html/date_time.html

Index

*Topic **package**

- bdtDd, 2
 - bdtDt, 3
 - bdtDu, 4
 - bdtPt, 5
 - bdtTz, 5
 - RcppBDT Date functions, 6
 - RcppBDT-constants, 7
 - RcppBDT-package, 2
- Apr (RcppBDT-constants), 7
- arith_bdtDd_bdtDd (bdtDd), 2
 - arith_bdtDd_bdtDt (bdtDd), 2
 - arith_bdtDd_int (bdtDd), 2
 - arith_bdtDt_bdtDd (bdtDt), 3
 - arith_bdtDt_int (bdtDt), 3
 - arith_bdtDu_bdtDu (bdtDu), 4
 - arith_bdtDu_bdtPt (bdtDu), 4
 - arith_bdtDu_int (bdtDu), 4
 - arith_bdtPt_bdtDu (bdtPt), 5
 - arith_bdtPt_double (bdtPt), 5
 - arith_double_bdtPt (bdtPt), 5
 - arith_int_bdtDd (bdtDd), 2
 - arith_int_bdtDt (bdtDt), 3
 - arith_int_bdtDu (bdtDu), 4
- Aug (RcppBDT-constants), 7
- bdt (bdtDt), 3
- bdtDd, 2
 - bdtDt, 3
 - bdtDu, 4
 - bdtPt, 5
 - bdtTz, 5
- compare_bdtDd_bdtDd (bdtDd), 2
 - compare_bdtDt_bdtDt (bdtDt), 3
 - compare_bdtDu_bdtDu (bdtDu), 4
 - compare_bdtPt_bdtPt (bdtPt), 5
- Date, 7
- days (bdtDd), 2
- Dec (RcppBDT-constants), 7
- Feb, 7
- Feb (RcppBDT-constants), 7
- fifth, 7
- fifth (RcppBDT-constants), 7
- first, 7
- first (RcppBDT-constants), 7
- format, Rcpp_bdtDd-method (bdtDd), 2
- format, Rcpp_bdtDt-method (bdtDt), 3
- format, Rcpp_bdtDu-method (bdtDu), 4
- format, Rcpp_bdtPt-method (bdtPt), 5
- format, Rcpp_bdtTz-method (bdtTz), 5
- fourth (RcppBDT-constants), 7
- Fri (RcppBDT-constants), 7
- getDay (RcppBDT Date functions), 6
- getDayOfWeek (RcppBDT Date functions), 6
- getDayOfYear (RcppBDT Date functions), 6
- getEndOfBizWeek, 3
- getEndOfBizWeek (RcppBDT Date functions), 6
- getEndOfMonth (RcppBDT Date functions), 6
- getFirstDayOfWeekAfter (RcppBDT Date functions), 6
- getFirstDayOfWeekInMonth (RcppBDT Date functions), 6
- getIMMDate (RcppBDT Date functions), 6
- getLastDayOfWeekBefore (RcppBDT Date functions), 6
- getLastDayOfWeekInMonth (RcppBDT Date functions), 6
- getMonth (RcppBDT Date functions), 6
- getNthDayOfWeek (RcppBDT Date functions), 6
- getYear (RcppBDT Date functions), 6
- hours (bdtDu), 4

Jan, [7](#)
Jan (RcppBDT-constants), [7](#)
Jul (RcppBDT-constants), [7](#)
Jun (RcppBDT-constants), [7](#)

Mar (RcppBDT-constants), [7](#)
May (RcppBDT-constants), [7](#)
microseconds (bdtDu), [4](#)
milliseconds (bdtDu), [4](#)
minutes (bdtDu), [4](#)
Mon, [7](#)
Mon (RcppBDT-constants), [7](#)

nanoseconds (bdtDu), [4](#)
Nov (RcppBDT-constants), [7](#)

Oct (RcppBDT-constants), [7](#)

Rcpp_bdtDd-class (bdtDd), [2](#)
Rcpp_bdtDt-class (bdtDt), [3](#)
Rcpp_bdtDu-class (bdtDu), [4](#)
Rcpp_bdtPt-class (bdtPt), [5](#)
Rcpp_bdtTz-class (bdtTz), [5](#)
RcppBDT (RcppBDT-package), [2](#)
RcppBDT Date functions, [6](#)
RcppBDT-constants, [7](#)
RcppBDT-package, [2](#)

Sat, [7](#)
Sat (RcppBDT-constants), [7](#)
second, [7](#)
second (RcppBDT-constants), [7](#)
seconds (bdtDu), [4](#)
Sep (RcppBDT-constants), [7](#)
show, Rcpp_bdtDd-method (bdtDd), [2](#)
show, Rcpp_bdtDt-method (bdtDt), [3](#)
show, Rcpp_bdtDu-method (bdtDu), [4](#)
show, Rcpp_bdtPt-method (bdtPt), [5](#)
show, Rcpp_bdtTz-method (bdtTz), [5](#)
Sun, [7](#)
Sun (RcppBDT-constants), [7](#)

third (RcppBDT-constants), [7](#)
Thu (RcppBDT-constants), [7](#)
Tue (RcppBDT-constants), [7](#)

Wed (RcppBDT-constants), [7](#)
weeks (bdtDd), [2](#)