

# Package ‘Rblpapi’

December 7, 2015

**Title** R Interface to Bloomberg

**Version** 0.3.2

**Date** 2015-12-06

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Author** Whit Armstrong, Dirk Eddelbuettel and John Laing

**Imports** Rcpp (>= 0.11.0), utils

**Suggests** fts, xts, zoo, knitr

**VignetteBuilder** knitr

**LazyLoad** yes

**LinkingTo** Rcpp, BH

**Description** An R Interface to Bloomberg is provided via the Blp API.

**SystemRequirements** A valid Bloomberg installation. API headers and dynamic library are downloaded from <<https://github.com/Rblp/blp>> during the build step. See <<http://www.bloomberglabs.com/api/>> for API documentation.

**License** file LICENSE

**RoxygenNote** 5.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-12-07 09:19:08

## R topics documented:

bdh . . . . .	2
bdp . . . . .	3
bds . . . . .	4
beqs . . . . .	5
blpAuthenticate . . . . .	6
blpConnect . . . . .	7
blpDisconnect . . . . .	8

defaultConnection . . . . .	8
fieldSearch . . . . .	9
getBars . . . . .	10
getMultipleTicks . . . . .	11
getTicks . . . . .	12
subscribe . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

bdh	<i>Run 'Bloomberg Data History' Queries</i>
-----	---------------------------------------------

---

## Description

This function uses the Bloomberg API to retrieve 'bdh' (Bloomberg Data History) queries

## Usage

```
bdh(securities, fields, start.date, end.date = NULL,
    include.non.trading.days = FALSE, options = NULL, overrides = NULL,
    identity = NULL, con = defaultConnection())
```

## Arguments

<code>securities</code>	A character vector with security symbols in Bloomberg notation.
<code>fields</code>	A character vector with Bloomberg query fields.
<code>start.date</code>	A Date variable with the query start date.
<code>end.date</code>	An optional Date variable with the query end date; if omitted the most recent available date is used.
<code>include.non.trading.days</code>	An optional logical variable indicating whether non-trading days should be included.
<code>options</code>	An optional named character vector with option values. Each field must have both a name (designating the option being set) as well as a value.
<code>overrides</code>	An optional named character vector with override values. Each field must have both a name (designating the override being set) as well as a value.
<code>identity</code>	An optional identity object.
<code>con</code>	A connection object as created by a <code>blpConnect</code> call, and retrieved via the internal function <code>defaultConnection</code> .

## Value

A list with as many entries as there are entries in `securities`; each list contains a `data.frame` with one row per observations and as many columns as entries in `fields`. If the list is of length one, it is collapsed into a single data frame.

**Author(s)**

Whit Armstrong and Dirk Eddelbuettel

**Examples**

```
## Not run:
bdh("SPY US Equity", c("PX_LAST", "VOLUME"), start.date=Sys.Date()-31)

## example for an options field: request monthly data; see section A.2.4 of
## http://www.bloomberglabs.com/content/uploads/sites/2/2014/07/blpapi-developers-guide-2.54.pdf
## for more
opt <- c("periodicitySelection"="MONTHLY")
bdh("SPY US Equity", c("PX_LAST", "VOLUME"),
    start.date=Sys.Date()-31*6, options=opt)

## End(Not run)
```

---

bdp

---

*Run 'Bloomberg Data Point' Queries*


---

**Description**

This function uses the Bloomberg API to retrieve 'bdp' (Bloomberg Data Point) queries

**Usage**

```
bdp(securities, fields, options = NULL, overrides = NULL, identity = NULL,
    con = defaultConnection())
```

**Arguments**

securities	A character vector with security symbols in Bloomberg notation.
fields	A character vector with Bloomberg query fields.
options	An optional named character vector with option values. Each field must have both a name (designating the option being set) as well as a value.
overrides	An optional named character vector with override values. Each field must have both a name (designating the override being set) as well as a value.
identity	An optional identity object.
con	A connection object as created by a <code>blpConnect</code> call, and retrieved via the internal function <code>defaultConnection</code> .

**Value**

A data frame with as many rows as entries in `securities` and columns as entries in `fields`.

**Author(s)**

Whit Armstrong and Dirk Eddelbuettel

**Examples**

```
## Not run:
  bdp(c("ESA Index", "SPY US Equity"), c("PX_LAST", "VOLUME"))

## using overrides (cf https://github.com/Rblp/Rblpapi/issues/67)
  bdp("EN00 Index", "MLI_OAS", overrides=c(MLI_DATE="20150831"))

## End(Not run)
```

---

 bds

---

*Run 'Bloomberg Data Set' Queries*


---

**Description**

This function uses the Bloomberg API to retrieve 'bds' (Bloomberg Data Set) queries

**Usage**

```
bds(securities, fields, options = NULL, overrides = NULL, identity = NULL,
    con = defaultConnection())
```

**Arguments**

<code>securities</code>	A character vector with security symbols in Bloomberg notation.
<code>fields</code>	A character string with a single Bloomberg query field.
<code>options</code>	An optional named character vector with option values. Each field must have both a name (designating the option being set) as well as a value.
<code>overrides</code>	An optional named character vector with override values. Each field must have both a name (designating the override being set) as well as a value.
<code>identity</code>	An optional identity object.
<code>con</code>	A connection object as created by a <code>blpConnect</code> call, and retrieved via the internal function <code>defaultConnection</code> .

**Value**

A list with as many entries as there are entries in `securities`; each list contains a data.frame with one row per observations and as many columns as entries in `fields`. If the list is of length one, it is collapsed into a single data frame.

**Author(s)**

Whit Armstrong and Dirk Eddelbuettel

**Examples**

```
## Not run:
  bds("GOOG US Equity", "TOP_20_HOLDERS_PUBLIC_FILINGS")

## End(Not run)
```

---

 beqs

*Run 'Bloomberg EQS' Queries*


---

**Description**

This function uses the Bloomberg API to retrieve 'beqs' (Bloomberg EQS Data) queries

**Usage**

```
beqs(screenName, screenType = "GLOBAL", language = "", group = "",
      date = NULL, verbose = FALSE, con = defaultConnection())
```

**Arguments**

screenName	A character string with the name of the screen to execute. It can be a user defined EQS screen or one of the Bloomberg Example screens on EQS
screenType	A character string of value PRIVATE or GLOBAL Use PRIVATE for user-defined EQS screen. Use GLOBAL for Bloomberg EQS screen.
language	An optional character string with the EQS language
group	An optional character string with the Screen folder name as defined in EQS
date	An optional Date object with the 'point in time' date of the screen to execute.
verbose	A boolean indicating whether verbose operation is desired, defaults to 'FALSE'.
con	A connection object as created by a blpConnect call, and retrieved via the internal function defaultConnection.

**Value**

A data frame object with the date in the first column and the requested EQS data in the remaining columns.

**Author(s)**

Rademeyer Vermaak and Dirk Eddelbuettel

## Examples

```
## Not run:
beqs("Global Oil Companies YTD Return")
beqs("Global Oil Companies YTD Return", "GLOBAL")
beqs("Global Oil Companies YTD Return", "GLOBAL", "GERMAN")
beqs("Global Oil Companies YTD Return", "GLOBAL", "GERMAN", "GENERAL")
beqs("Global Oil Companies YTD Return", "GLOBAL", "ENGLISH", "GENERAL", as.Date("2015-09-30"))

## End(Not run)
```

---

blpAuthenticate	<i>Authenticate Bloomberg API access</i>
-----------------	------------------------------------------

---

## Description

This function authenticates against the the Bloomberg API

## Usage

```
blpAuthenticate(uuid, host = "localhost", ip.address,
  con = defaultConnection())
```

## Arguments

uuid	A character variable with a unique user id token
host	A character variable with a hostname, defaults to 'localhost'
ip.address	An optional character variable with an IP address
con	A connection object as created by a blpConnect call, and retrieved via the internal function defaultConnection.

## Value

Not sure. May just be the side effect of having the session authenticated.

## Author(s)

Whit Armstrong and Dirk Eddelbuettel

---

blpConnect	<i>Establish connection to Bloomberg service</i>
------------	--------------------------------------------------

---

### Description

This function connects to the Bloomberg API

### Usage

```
blpConnect(host = getOption("blpHost", "localhost"),
           port = getOption("blpPort", 8194L), default = TRUE)
```

### Arguments

host	A character option with either a machine name that is resolvable by DNS, or an IP address. Defaults to 'localhost'.
port	An integer variable with the connection port. Default to 8194L.
default	A logical indicating whether this connection should be saved as the default, as opposed to returned to the user. Default to TRUE.

### Details

For both host and port argument, default values can also be specified via [options](#) using, respectively, the named entries blpHost and blpConnect.

If an additional option blpAutoConnect is set to 'TRUE', a connection is established in the `.onAttach()` function and stored in the package environment. This effectively frees users from having to explicitly create such an object.

### Value

In the default=TRUE case nothing is returned, and this connection is automatically used for all future calls which omit the con argument. Otherwise a connection object is returned which is required by all the accessor functions in the package.

### Author(s)

Whit Armstrong and Dirk Eddelbuettel

### Examples

```
## Not run:
  con <- blpConnect() # adjust as needed

## End(Not run)
```

---

blpDisconnect	<i>Placeholder function for disconnection from Bloomberg</i>
---------------	--------------------------------------------------------------

---

**Description**

This function provides an empty stub and does not really disconnect.

**Usage**

```
blpDisconnect(con)
```

**Arguments**

con	A connection object
-----	---------------------

**Details**

The internal connection object is managed via finalizers. As such the connection is only destroyed, and the connection removed, once the packaged is unloaded or the session is otherwise terminated.

**Value**

A boolean is returned; it simply states whether the connection object was small or large relative to an arbitrary cutoff of 1000 bytes.

**Author(s)**

Whit Armstrong and Dirk Eddelbuettel

**Examples**

```
## Not run:  
  blpDisconnect(con)  
  
## End(Not run)
```

---

defaultConnection	<i>Return the default connection object</i>
-------------------	---------------------------------------------

---

**Description**

This function return the default connection object from the package environment. If no default connection has been established yet, an error message is shown,

**Usage**

```
defaultConnection()
```



**Details**

For the connection object, the required arguments `host` and `port` argument can be set via [options](#). In addition, if an additional option `blpAutoConnect` is set to 'TRUE', a connection is established in the `.onAttach()` function and stored in the package environment. This effectively frees users from having to explicitly create such an object. Of course, the user can also call `blpConnect` explicitly and store the connection object. This helper function looks up the stored connection object and returns it. In case no connection has been established, an error message is shown.

**Author(s)**

Whit Armstrong and Dirk Eddelbuettel

**Examples**

```
## Not run:
con <- defaultConnection()

## End(Not run)
```

---

fieldSearch

*Search for matching data fields*

---

**Description**

This function searches for matching Bloomberg data fields given a search term.

**Usage**

```
fieldSearch(searchterm, excludeterm = "Static", con = defaultConnection())
```

**Arguments**

<code>searchterm</code>	A string with the term to search for
<code>excludeterm</code>	A string with an expression for matches to excludes, defaults to "Static"
<code>con</code>	A connection object as created by a <code>blpConnect</code> call, and retrieved via the internal function <code>defaultConnection</code> .

**Value**

A data.frame with three columns of the id, mnemonic and description of each match.

**Author(s)**

Dirk Eddelbuettel

**Examples**

```
## Not run:
con <- blpConnect()
res <- fieldSearch("vwap")

## End(Not run)
```

getBars

*Get Open/High/Low/Close/Volume Bars from Bloomberg***Description**

This function uses the Bloomberg API to retrieve bars for the requested security.

**Usage**

```
getBars(security, eventType = "TRADE", barInterval = 60,
        startTime = Sys.time() - 60 * 60 * 6, endTime = Sys.time(),
        options = NULL, verbose = FALSE, returnAs = getOption("blpType",
        "matrix"), tz = Sys.getenv("TZ", unset = "UTC"),
        con = defaultConnection())
```

**Arguments**

security	A character variable describing a valid security ticker
eventType	A character variable describing an event type; default is 'TRADE'
barInterval	A integer denoting the number of minutes for each bar
startTime	A Datetime object with the start time, defaults to one hour before current time
endTime	A Datetime object with the end time, defaults to current time
options	An optional named character vector with option values. Each field must have both a name (designating the option being set) as well as a value.
verbose	A boolean indicating whether verbose operation is desired, defaults to 'FALSE'
returnAs	A character variable describing the type of return object; currently supported are 'matrix' (also the default), 'fts', 'xts' and 'zoo'
tz	A character variable with the desired local timezone, defaulting to the value 'TZ' environment variable, and 'UTC' if unset
con	A connection object as created by a blpConnect call, and retrieved via the internal function defaultConnection.

**Value**

A numeric matrix with elements 'time' (as a 'POSIXct' object), 'open', 'high', 'low', 'close', 'numEvents', 'volume', 'value' or an object of the type selected in returnAs. Note that the 'time' value is adjusted: Bloomberg returns the *opening* time of the bar interval, whereas financial studies typically refer to the most recent timestamp. For this reason we add the length of the bar interval to time value from Bloomberg to obtain the time at the end of the interval.

**Author(s)**

Dirk Eddelbuettel

**Examples**

```
## Not run:
  getBars("ES1 Index")

## End(Not run)
```

---

```
getMultipleTicks      Get Multiple Ticks from Bloomberg
```

---

**Description**

This function uses the Bloomberg API to retrieve multiple ticks for the requested security.

**Usage**

```
getMultipleTicks(security, eventType = c("TRADE", "BID", "ASK"),
  startTime = Sys.time() - 60 * 60, endTime = Sys.time(), verbose = FALSE,
  returnAs = getOption("blpType", "matrix"), tz = Sys.getenv("TZ", unset =
    "UTC"), con = defaultConnection())
```

**Arguments**

security	A character variable describing a valid security ticker
eventType	A character vector describing event types, default is c("TRADE", "BID", "ASK")
startTime	A Datetime object with the start time, defaults to one hour before current time
endTime	A Datetime object with the end time, defaults to current time
verbose	A boolean indicating whether verbose operation is desired, defaults to 'FALSE'
returnAs	A character variable describing the type of return object; the default is return a matrix with results as received; optionally a 'wide' xts object with merged data can be returned
tz	A character variable with the desired local timezone, defaulting to the value 'TZ' environment variable, and 'UTC' if unset
con	A connection object as created by a blpConnect call, and retrieved via the internal function defaultConnection.

**Value**

A numeric matrix with elements 'time', (as a 'POSIXct' object), 'values' and 'sizes', or an object of the type selected in returnAs.

**Author(s)**

Dirk Eddelbuettel

---

 getTicks

*Get Ticks from Bloomberg*


---

**Description**

This function uses the Bloomberg API to retrieve ticks for the requested security.

**Usage**

```
getTicks(security, eventType = "TRADE", startTime = Sys.time() - 60 * 60,
         endTime = Sys.time(), verbose = FALSE, returnAs = getOption("blpType",
         "matrix"), tz = Sys.getenv("TZ", unset = "UTC"),
         con = defaultConnection())
```

**Arguments**

security	A character variable describing a valid security ticker
eventType	A character variable describing an event, default is 'TRADE'.
startTime	A Datetime object with the start time, defaults to one hour before current time
endTime	A Datetime object with the end time, defaults to current time
verbose	A boolean indicating whether verbose operation is desired, defaults to 'FALSE'
returnAs	A character variable describing the type of return object; currently supported are 'matrix' (also the default), 'fts', 'xts' and 'zoo'
tz	A character variable with the desired local timezone, defaulting to the value 'TZ' environment variable, and 'UTC' if unset
con	A connection object as created by a blpConnect call, and retrieved via the internal function defaultConnection.

**Value**

A numeric matrix with elements 'time', (as a 'POSIXct' object), 'values' and 'sizes', or an object of the type selected in returnAs.

**Author(s)**

Dirk Eddelbuettel

**Examples**

```
## Not run:
  getTicks("ES1 Index")

## End(Not run)
```

---

subscribe	<i>Subscribe to streaming market data</i>
-----------	-------------------------------------------

---

### Description

This function uses the Bloomberg API to stream live market data

### Usage

```
subscribe(securities, fields, fun, options = NULL, identity = NULL,  
          con = defaultConnection())
```

### Arguments

securities	A character vector with security symbols in Bloomberg notation.
fields	A character vector with Bloomberg query fields.
fun	An R function to be called on the subscription data.
options	An optional named character vector with option values. Each field must have both a name (designating the option being set) as well as a value.
identity	An optional identity object.
con	A connection object as created by a <code>blpConnect</code> call, and retrieved via the internal function <code>defaultConnection</code> .

### Details

The `subscribe` function allows one to subscribe to streaming market quotes.

Full details of the subscription string can be found in the header file [blpapi\\_subscriptionlist.h](#).

### Value

This function always returns `NULL`.

### Author(s)

Whit Armstrong

### References

<http://static.bloomberglabs.com/api/cpp/doc/3.8>

### Examples

```
## Not run:  
  subscribe(securities=c("TYZ5 Comdty", "/cusip/912810RE0@BGN"),  
            fields=c("LAST_PRICE", "BID", "ASK"),  
            fun=function(x) print(str(x$data)))  
  
## End(Not run)
```

# Index

bdh, [2](#)  
bdp, [3](#)  
bds, [4](#)  
beqs, [5](#)  
blpAuthenticate, [6](#)  
blpConnect, [7](#)  
blpDisconnect, [8](#)  
  
defaultConnection, [8](#)  
  
fieldSearch, [9](#)  
  
getBars, [10](#)  
getMultipleTicks, [11](#)  
getTicks, [12](#)  
  
options, [7, 9](#)  
  
subscribe, [13](#)