

Package ‘VhayuR’

January 2, 2012

Version 1.1.2

Date 2008-08-02

Title Vhayu R Interface

Author The Brookhaven Group

Maintainer The Brookhaven Group <info@thebrookhavengroup.com>

Description Vhayu R Interface

SystemRequirements Vhayu

Depends chron, zoo

Imports utils, stats

LazyLoad no

License GPL-2

OS_type windows

Repository CRAN

Date/Publication 2010-05-08 10:04:41

R topics documented:

VhayuR-package	2
vh.flexrecdef	3
vh.get.data.frame	4
vh.options	6

Index	8
--------------	----------

Description

R interface to Vhayu time series database

Details

VhayuR allows one to read the VhayuR Velocity time series database from within R. The package utilizes lower level dynamic load libraries provided by Vhayu. These dynamic load libraries are necessary to run this package with Vhayu and must be separately obtained from Vhayu and installed. (It is also possible to run the package in a demo mode with files simulating Vhayu and in that mode the Vhayu libraries are not required.)

Objects are retrieved from the Vhayu data base as `data.frame` or `zoo` class R objects. The class of the returned time component may also be specified or is `POSIXct` if not specified.

In "demo" mode files are used in place of Vhayu. If care is taken with option settings then it may be possible to simulate a Vhayu session using files such that only the options change between the Vhayu and file-based sessions. This can be useful for demonstrations where Vhayu is not available.

The primary user functions are `vh.get.data.frame` and `vh.get.zoo`.

There is an option system that can be used to set the server. It can also be used to set the default arguments of `vh.get.data.frame` and `vh.get.zoo`. Other options are also available. The option system is very similar to the option system in R but stores its options in the VhayuR package. See `vh.options`.

The first time the user attempts to access Vhayu in any session VhayuR automatically loads the required dynamic load library (dll) and connects. Thus the only explicit tasks the user must do is install the dynamic load libraries from Vhayu on the machine (which is a one-time operation) and set the server IP address. The server must be set each session but this can be automated by placing the set server instructions in your R `Startup` file which is typically `Rprofile.site` found in the `etc` subdirectory of your R distribution. The examples below illustrate this.

Examples can be found below, in the accompanying demo accessible via the R `demo` command and in further examples in the help files.

References

Vahyu Velocity Vhayu R Integration Guide. Vhayu Velocity API Reference Guide.

Examples

```
## Not run:  
  
# change this to your server IP address  
# (You don't need this if you set up your startup file  
# appropriately as discussed below.)  
vh.options(server = "10.10.1.50")
```

```
str(vh.options()) # show options

vh.flexrecdef() # check available flex record definitions
vh.flexrecdef("VhTrade") # view definition for VhTrade

goog.df <- vh.get.data.frame("GOOG",
  frDef = "VhTrade",
  startTime = "20051201 09:30:00",
  endTime = "20051201 16:00:00",
  fieldNames = "VhExchgTime VhPrice",
  FUN = as.chron)

### This R command gives the location of your Rprofile.site file
file.path(R.home(), "etc", "Rprofile.site")

### Place these lines in your R Startup file
### Replace IP address shown with the one you use to connect to Vhayu.
setHook(packageEvent("VhayuR", "attach"),
  function(...) vh.options(server = "10.10.1.50")
)

## End(Not run)
```

vh.flexrecdef

Vhayu Flex Record Definitions

Description

Discover and list Vhayu Flex records and their definitions.

Usage

```
vh.flexrecdef(x = NULL)
```

Arguments

x NULL or character vector

Value

If x is NULL return a character vector of flex record definition names; otherwise, return an unformatted flex record definition as a character vector.

Examples

```
## Not run:
vh.flexrecdef()
vh.flexrecdef("VhTrade")

## End(Not run)
```

vh.get.data.frame	<i>Get data from Vhayu.</i>
-------------------	-----------------------------

Description

Retrieve data from Vhayu server (or file if in demo mode).

Usage

```
vh.get.data.frame(x, ...)
vh.get.zoo(x, ...)
```

Arguments

x	Name of Vhayu table (or if in demo mode then the name or the name and filepath of the file).
...	Other arguments.

Details

If the Vhayu option "demo" is FALSE, which it is by default, then these functions retrieve data from the Vhayu server. In that case the arguments "frDef", "startTime", "endTime", "fieldNames", "filter" and "maxRows" may be specified or will be taken from the Vhayu options if not specified. As discussed in the Vhayu R Integration Guide, "frDef" is the name of a flex record definition (see the command [vh.flexrecdef](#) which can display the possible names usable here), "startTime" and "endTime" are each strings of the form "yyyymmdd HH:MM:SS", filter is the name of a Vhayu filter and maxRows is the maximum number of rows to retrieve or 0 if all records are desired.

If the Vhayu option "demo" is TRUE then `vh.get.zoo` will call [read.zoo](#) instead of Vhayu and `vh.get.data.frame` will call [read.table](#). Any arguments known to these two functions may be passed while the remaining arguments will be ignored. The x argument is interpreted as a file name which is looked up along the path defined by the Vhayu option "datapath". If not found it will also search a file of the same name suffixed with ".csv", ".txt" or ".dat". If a pathname and filename are specified rather than just a filename then that path is used. If the [read.zoo](#) and [read.table](#) arguments of "sep" or "header" are not specified then there will be an attempt to infer them from the first line of the file assuming a file with a comma in the first line should have sep = "," and a file with an alphabetic in its first line should have header = TRUE.

In the case of `vh.get.zoo` the aggregate argument can be used to specify an R function such as `vh.tail1`, `mean` or `median` to aggregate data values corresponding to a single time value. If aggregate is not specified and if there are multiple values for any time then a warning will be issued; however, a zoo object will still be produced. Such an object may not participate in merges but may be printed, plotted or aggregated. In the case of `vh.get.data.frame` aggregate is not used and these comments do not apply.

Value

vh.get.data.frame returns a data frame and vh.get.zoo returns a zoo object. The times are returned as POSIXct objects unless the FUN argument is specified. In this latter case, FUN is an R function that converts times from POSIXct to whatever class is desired. See [DateTimeClasses](#).

References

Vahyu Velocity Vhayu R Integration Guide. Vhayu Velocity API Reference Guide.

Examples

```
## Not run:
vh.flexrecdef() # check which flex record definitions are available
vh.flexrecdef("VhTrade") # view definition for VhTrade

goog.df <- vh.get.data.frame("GOOG",
  frDef = "VhTrade",
  startTime = "20051201 09:30:00",
  endTime = "20051201 16:00:00",
  fieldNames = "VhExchgTime VhPrice",
  FUN = as.chron)

goog.zoo <- vh.get.zoo("GOOG",
  frDef = "VhTrade",
  startTime = "20051201 09:30:00",
  endTime = "20051201 16:00:00",
  fieldNames = "VhExchgTime VhPrice",
  FUN = as.chron,
  aggregate = vh.tail1)

## same but set arguments through options
## We have also set the format option which is not application and ignored.

vh.options(frDef = "VhTrade",
  startTime = "20051201 09:30:00",
  endTime = "20051201 16:00:00",
  fieldNames = "VhExchgTime VhPrice",
  FUN = as.chron,
  aggregate = vh.tail1,
  format = "%Y%m%d")

# Assume there is a csv formatted file called GOOG.csv
# with yyyyymmdd dates and assume the vh.options is still as above.
# Since only options format, FUN and aggregate
# apply to read.zoo only those will be used.
# Thus by setting options appropriate the same script may work
# with files and Vhayu.
vh.options(demo = TRUE)
goog.file <- goog <- vh.get.zoo("GOOG")
```

```
## End(Not run)
```

 vh.options

Vhayu Options

Description

Options in the VhayuR package.

Usage

```
vh.defaultOptions(set = c("all", "set", "unset"))
vh.getOption(name)
vh.options(...)
```

Arguments

set	Return all default options, just those which are set to a non-NULL value in the Vhayu options or just those which are currently NULL.
name	The name of a Vhayu option.
...	See below.

Details

Vhayu options are like R options except they are stored in the VhayuR package and not with regular R options. `vh.options` is similar to the R `options` command and `vh.getOption` is similar to the R `getOption` command.

Invoking `vh.options()` with no arguments returns a list with the current values of the options. Note that not all options listed below are set initially. To access the value of a single option, one should use `getOption("demo")`, e.g., rather than `options("demo")` which is a *list* of length one. `vh.defaultOptions()` returns a named list of the default, rather than current, values of all options (or of just the default options whose current values are set or not set).

Each time the VhayuR package is loaded, a new empty list is defined to hold the options and the default values are loaded into that list.

Value

"`vh.options`" and "`vh.defaultOptions`" each returns a named list. "`vh.getOption`" returns the value of a single Vhayu option.

Options used in VhayuR package

loaded: TRUE if the VhayuR.dll dynamic load library (dll) is loaded. Automatically set. Not normally changed by user.

port: This option is the port address that is used to contact the Vhayu server. Usually the default is acceptable so the user normally need not to modify this value.

server: This option should be set by the user to the IP address of their Vhayu server during each session.

verbose: This option provides additional messages when running the VhayuR package. When the VhayuR package is loaded verbose is loaded with the value of the R option of the same name.

Other options are discussed in [vh.get.data.frame](#).

See Also

[options](#)

Examples

```
# compactly list Vhayu options
str(vh.options())

# set an option and then restore to original state
vhgetOption("x")
op <- vh.options(x = 4)
vhgetOption("x")
vh.options(op)
vhgetOption("x")

## Not run:
# reset all options which have defaults to their defaults
vh.options(vh.defaultOptions())
# reset all options which have defaults and are unset
vh.options(vh.defaultOptions("unset"))

vh.options(server = "10.10.1.50")

## End(Not run)
```

Index

*Topic **ts**

- vh.flexrecdef, 3
- vh.get.data.frame, 4
- vh.options, 6
- VhayuR-package, 2

DateTimeClasses, 5
demo, 2

getOption, 6

options, 6, 7

read.table, 4
read.zoo, 4

Startup, 2

- vh.defaultOptions (vh.options), 6
- vh.flexrecdef, 3, 4
- vh.get.data.frame, 2, 4, 7
- vh.get.zoo, 2
- vh.get.zoo (vh.get.data.frame), 4
- vh.getOption (vh.options), 6
- vh.options, 2, 6
- vh.tail1 (vh.get.data.frame), 4
- VhayuR-package, 2