Package ‘padi’

October 4, 2006

Title PADI

Description Protocol for Application Database Interface

Depends R (>= 2.0.0), syskern (>= 2006.10-1), dsepadi, tframe (>= 2006.10-1)

Enhances dsepadi

Version 2006.10-1

Date 2006-10-04

LazyLoad yes

License Free. See the LICENCE file for details.

Author Paul Gilbert <pgilbert@bank-banque-canada.ca>

Maintainer Paul Gilbert <pgilbert@bank-banque-canada.ca>

URL http://www.bank-banque-canada.ca/pgilbert

R topics documented:

PADIserverProcess *PADI Functions*

Description

see details

Usage

```
PADIserverProcess()
PADIcleanupScript()
startPADIserver(server=Sys.info()[["nodename"]],
    server.process=PADIserverProcess(), dbname=NULL)
cleanupPADIserver(process, cleanup.script=PADIcleanupScript())
killPADIserver(kill.script="killserver",
    server.process=PADIserverProcess())
```

Arguments

server	string or vector of strings indicating the server where the series will be found. Scalar values are expanded to a vector of appropriate length.
dbname	string or vector of strings indicating additional information for the server. Scalar values are expanded to a vector of appropriate length.
server.process	string indicating the name of the process to be used to start a server process.
cleanup.script	string indicating the name of the process to be used to shut down a server process.
process	argument to cleanup.script, not actually the process ID.
kill.script	string indicating the name of the process to be used to kill a server process.

Details

The function startPADIserver uses a script to start a server. The function cleanupPADIserver uses the information returned by startPADIserver to terminate the server. The function killPADIserver looks for a server and kills it. The default scripts for starting and stopping the server are determined by the functions PADIserverProcess and PADIcleanupScript, which use the Unix environment variables PADI, PADI_STARTUP and PADI_CLEANUP. Scripts for starting and stopping the server are only relevant in cases when it is necessary to start or stop a server.

Value

depends

See Also

getpadi putpadi

checkPADIserver *PADI Functions*

Description

see details

Usage

```
checkPADIserver(server=Sys.info()[["nodename"]],
      user=Sys.info()[["user"]], timeout=50)
```

Arguments

server	string indicating the network name of the server."
user	optional string indicating user ID used by the server to start the process."
timeout	an integer indicating the number of seconds to wait before concluding that the server is not available.

Details

The function load.padi loads the associated compiled C object code. The function checkPADIserver checks if a server is running.

Value

depends

See Also

getpadi putpadi getpadi.default putpadi.default

getfm *Function is for backwards compatability at the BOC*

Description

Function is for backwards compatability at the BOC

Usage

```
getfm(dbname,series, starty=0,startp=0, endy=0,endp=0,
   nobs=0,max.obs=2000, transformations=NULL, pad=FALSE)
putfm(data, dbname, seriesNames)
```

Arguments

dbname	
series	see getpadi.
starty	see getpadi.
startp	start period.
endy	see getpadi.
endp	end period.
nobs	see getpadi.
max.obs	see getpadi.
transformations	see getpadi.
pad	see getpadi.
data	a object with data to put on the server.
seriesNames	vector of strings to use for series identifiers.

Details

Depreciated.

Value

see getpadi.

See Also

getpadi.default getpadi

---

getpadi.default *Get Data from TSPADI Database Interface*

Description

Get data from a TSPADI database interface.

Usage

```
## Default S3 method:
getpadi(series, server=Sys.info()[["nodename"]], dbname="",
            start.server=TRUE, server.process=PADIserverProcess(),
            cleanup.script=PADIcleanupScript(),
            starty=0,startm=0,startd=1, endy=0,endm=0,endd=1,
            nobs=0,max.obs=2000, transformations=NULL, pad=FALSE,
            user=Sys.info()[["user"]], passwd="",
            stop.on.error=TRUE, use.tframe=FALSE, warn=TRUE, timeout=60)
```

Arguments

`series`	A character string giving the name of the series. Alternately, series can be a vector of character strings specifying multiple series.
`server`	A character string giving the network name of the server which is to be requested to supply the series. If series specifies multiple series and they are not all on the same server then server should be a vector of character strings with elements corresponding to the elements of series.
`dbname`	A character string giving additional information to the server about the location of the series (eg. the name of a database). If series specifies multiple series and they are not all on the same database then dbname should be a vector of character strings with elements corresponding to the elements of series.
`starty`	An integer indicating the starting year.
`startm`	An integer indicating the starting period.
`startd`	An integer indicating the starting day.
`endy`	An integer indicating the ending year.
`endm`	An integer indicating the ending period.
`endd`	An integer indicating the ending day.
`nobs`	The number of observations.
`max.obs`	integer indicating the possible returned data size used to define the size of the buffer to prepare.
`transformations`	A character string giving transformations to be applied to the series (e.g. "log"). If multiple series are being requested then transformations can be a single string, in which case it is applied to all series, or a vector of character strings, one for each series. If no transformation is to be applied to some series then "" should be used.
`pad`	If FALSE (default) then all series are truncated to the interection of available time periods (i.e. the latest start date and earliest end date). If TRUE then series are padded with NA so the result starts at the earliest available observation and ends at the last available observation.
`start.server`	try to start a server if one is not running
`server.process`	command to execute in an attempt to start a server
`cleanup.script`	command to execute to terminate a server if one is started
`user`	user id for access to the database (if necessary)
`passwd`	password for access to the database (if necessary)
`stop.on.error`	If TRUE then stop is executed when an error occurs. Otherwise, the error message is returned and the calling program must deal with it.
`use.tframe`	If use.tframe=FALSE then ts() is used to construct the time series, otherwise the tframe utilities are used. Certain transformations available with DSE require the tframe stucture and an error may result if these transformations are attempted with use.tframe=FALSE.
`warn`	Print warning messages for some crude frequency conversions (weekly data).
`timeout`	an integer indicating the number of seconds to wait before concluding that the server is not available.

Details

The function getpadi retrieves data from a (time series) server. The padi code is also available (including the PADI server side) at www.bank-banque-canada.ca/pgilbert.

(This documentation could use some cleaning up, but first I am really hoping to convert the whole interface to a more modern, non-RPC based, mechanism.)

start. and end dates or start dates and nobs can be supplied. If all are set to zero (the default) then all data is retrieved, provided max.obs is large enough. If more than one series is to be returned then series should be a vector of strings. In this case dbname must be a vector of corresponding length or a single string which is applied to all series. All series must have the same frequency. If the number of observations in any single series is larger than max.obs then an error will occur and max.obs should be set larger. For many data frequencies startd and endd can be omitted.

If start.server is FALSE then there will be no attempt to start a server and the function will stop if a server is not running. If start.server is TRUE (the default) then if server==Sys.info()[["nodename"]] (the default) and a server is not already running, there will be an attempt to start a server using the argument server.process. The default is determined by PADIserverProcess()

The argument server.process is only used if it is necessary to start a server.

If specified, server would typically be a single string, though there is some attempt to handle vectors of strings (indicating different servers for each series).

The string dbname is passed to the server, but it may or may not be used, depending on the server implementation. If the server is being started then it will run in the Unix pwd and local path names should work, BUT in general there is no guarantee that the server is running in the pwd and complete path names may be required.

Start and end dates or start dates and nobs can be supplied. If all are set to zero (the default) then all data is retrieved, provided max.obs is large enough.

If the starting date and ending date and number of observations is set to zero, it will return the whole series. If the starting and ending dates are zero but number of observations is set, the LAST numobs observations are returned. If the starting date is sent, ending date is zero and numobs is non-zero, the FIRST numobs starting from start date are returned.

The size of the data array passed to C is the larger of nobs and max.obs. A ts matrix is returned. If more than one series is to be returned then series should be a vector of strings. In this case server must be a vector of corresponding length or a single string which is applied to all series. All series must have the same frequency and the time window is the intersection of the window for each series (i.e. the latest start and earliest end date). startd and endd provide for a tag (day).

If the starting date and ending date and number of observations is set to zero, it will return the whole series. If the starting and ending dates are zero but number of observations is set, the LAST numobs observations are returned. If the starting date is sent, ending date is zero and numobs is non-zero, the FIRST numobs starting from start date are returned.

If transformations is not null it should be a vector of strings, one for each series (with "" for any series which is not to be transformed), or a single string which is applied to each series. The transformations are applied by eval(call(transformations[i], .).

If use.tframe=FALSE then ts() is used to construct the time series, otherwise the tframe approach is used. (See the tframe package.) Certain transformations available with DSE require the tframe structure and an error may result if these transformations are attempted with use.tframe=FALSE.

If stop.on.error is TRUE then stop is executed if a data retrieval error occurs. If stop.on.error is FALSE then the error message is returned, which means the calling function will need to handle the error.

If warn is TRUE then warnings are issued for certain data conversions (weekly data) which may not work in all case. If warn is these warnings are not issued. timeout is the period to wait (in seconds) before generating an error.

Value

A time series matrix with a column for each series.

See Also

`checkPADIserver` `putpadi.default` `getpadi` `putpadi`

Examples

```
if(require("padi") && checkPADIserver("ets")){
  cpi <-getpadi("P100000", server= "ets")# June 1992=100
  cpi <-getpadi("P100000", starty=1988, startm=1,endy=1990, endm=12, server= "ets")
  cpi <-getpadi( "P100000",starty=1988, startm=1,nobs=8, server= "ets")
  data <- getpadi( c("V122491","V37124","P100000"), server= "ets" )

  cpi <-getpadi("P100000", server= "ets", use.tframe=TRUE)
  data <- getpadi( c("V122491","V37124","P100000"), server= "ets", use.tframe=TRUE)
 }
```

`putpadi.default` *PADI Functions*

Description

see details

Usage

```
## Default S3 method:
putpadi(data,  server=Sys.info()[["nodename"]], dbname="",
                 series=seriesNames(data),
                 start.server=TRUE, server.process=PADIserverProcess(),
                 cleanup.script=PADIcleanupScript(),
                 user=Sys.info()[["user"]], passwd="",
                 stop.on.error=TRUE, warn=TRUE, timeout=60)
```

Arguments

`data`	a tfPADIdata object with data to put on the server.
`server`	string or vector of strings indicating the server where the series will be found. Scalar values are expanded to a vector of appropriate length.
`dbname`	string or vector of strings indicating additional information for the server. Scalar values are expanded to a vector of appropriate length.
`series`	vector of strings to use for series identifiers.
`start.server`	logical indicating if a (local) server should be started.
`server.process`	string indicating the name to be used to start a server process.

`cleanup.script`
: string indicating the name to be used to shut down a server process.

`user`
: an optional string used by the server to check permission.

`passwd`
: an optional string used by the server to check permission.

`stop.on.error`
: logical indicating if the function should stop if any series produces an error, or continue with other series.

`warn`
: logical indicating if warning messages should be supressed.

`timeout`
: an integer indicating the number of seconds to wait before concluding that the server is not available.

Details

The function putpadi writes data to a specified databases on a specified server (default local).

If start.server is FALSE then there will be no attempt to start a server and the function will stop if a server is not running. If start.server is TRUE (the default) then if server==Sys.info()[["nodename"]] (the default) and a server is not already running, there will be an attempt to start a server using the argument server.process. The default is a "local mode" Fame server.

The argument server.process is only used if it is necessary to start a server.

If specified, server would typically be a single string, though there is some attempt to handle vectors of strings (indicating different servers for each series).

dbname should typically be supplied. (but some servers might accept an empty string ("") or ignore any string supplied. If the server is being started then it will run in the Unix pwd and local path names should work, BUT in general there is no guarantee that the server is running in the pwd and complete path names may be required. data can be a vector, matrix, time series or time series matrix. Dates are used when available. If data is a vector or single time series, then server, dbname, and series should have length 1. If data is a matrix or time series matrix with more than one series then server, dbname, and series should be character strings of the same length or, server and dbname can be length 1, in which case the string will be repeated for all series. If they do not exist then an error is indicated. The strings user and passwd are passed to the server, but may or may not be used, depending on the server implementation. If stop.on.error is TRUE (default) the function stops if there is an error writing any series. Otherwise, the result is TRUE or FALSE for each series, indicating success or failure.

Value

depends

See Also

`checkPADIserver` `putpadi` `getpadi` `getpadi.default` `putpadi.default`

Index