Package ‘stacomirtools’

February 20, 2015

Type Package
Title stacomi ODBC connection class
Version 0.3
Date 2013-01-07
Author Cedric Briand
Maintainer Cedric Briand<cedric.briand00@gmail.com>
Description S4 class wrappers for ODBC connection.
License GPL (>= 2)
Collate 'ConnectionODBC.r' 'RequeteODBC.r' 'RequeteODBCwhere.r'
'RequeteODBCwheredate.r' 'utilitaires.r'
LazyLoad yes
Depends RODBC,xtable,methods
Repository CRAN
Repository/R-Forge/Project stacomir
Repository/R-Forge/Revision 113
Repository/R-Forge/DateTimeStamp 2013-01-07 09:14:52
Date/Publication 2013-01-07 17:56:02
NeedsCompilation no

R topics documented:

  stacomirtools-package ........................................... 2
  chnames ........................................................... 2
  connect-methods .................................................. 3
  ConnectionODBC-class ........................................... 4
  ex ................................................................. 5
  funhtml ............................................................ 5
  induk ............................................................... 6
  is.even ............................................................ 6
  is.odd .............................................................. 7
Description

This package contains S4 wrappers for ODBC connection and some utilities

Details

Package: stacomirtools
Type: Package
Version: 0.3
Date: 2012-12-23
License: GPL (>= 2)
LazyLoad: yes

Author(s)

Cedric Briand <cedric.briand@epcb-vilaine.fr>

chnames

This function replaces the variable names in a data.frame

Usage

chnames(objet, old_variable_name, new_variable_name)
Arguments

objet a data frame
old_variable_name a character vector with old variables names
new_variable_name a character vector with new variables names

Value

objet

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Methods

signature(objet = ConnectionODBC) connect an ODBC database, and eventually leaves it open for further queries, the connection may send a message in the native language if stacomIR package is in use
signature(objet = RequeteODBC) connect an ODBC database, performs an SQL request
signature(objet = RequeteODBCwhere) connect an ODBC database, performs an SQL request with where clause
signature(objet = RequeteODBCwheredate) connect an ODBC database, performs an SQL request with where clause for an interval

Examples

showMethods("connect")
##
#objet<-new("RequeteODBCwhere")
#connect(objet)
Description
Mother class for connection, opens the connection but does not shut it

Objects from the Class
Objects can be created by calls of the form new("ConnectionODBC", ...).

- `baseODBC`: Object of class "vector" The database
- `silent`: Object of class "logical" The mode
- `etat`: Object of class "character" The state
- `connection`: Object of class "ANY" The connection

Slots
- `baseODBC`: Object of class "vector" The database
- `silent`: Object of class "logical" The mode
- `etat`: Object of class "character" The state
- `connection`: Object of class "ANY" The connection

Methods
- `connect` signature(objet = "ConnectionODBC"): Connection to the database

Note
Opens the connection but does not close it. This function is intended to be used with stacomir package, where the error message are collected from the database. It has also been programmed to work without the stacomir package, as it will test for the existence of envir_stacomir environment.

Author(s)
cedric.briand@eptb-vilaine.fr

Examples
```
showClass("ConnectionODBC")
## Not run:
# this is the mother class, you don't have to use it, please use requeteODBC and daughter class instead
objet <- new("ConnectionODBC")
objet@baseODBC <- c("myODBCconnection","myusername","mypassword")
objet@silent <- FALSE
objet <- connect(objet)
odbcClose(objet@connection)

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

ex fonction to write to the clipboard

### Usage

```r
ex(d = NULL)
```

### Arguments

- **d**
  - a dataframe

### Author(s)

Cedric Briand <cedric.briand"at"eptb-vilaine.fr>

---

### Description

funhtml function used to print the html tables of output (see xtable documentation)

### Usage

```r
funhtml(data, caption=NULL, top=TRUE, outfile=NULL, clipboard=FALSE, append=TRUE, digits=NULL,...)
```

### Arguments

- **data**
  - a data frame
- **caption**
  - the caption
- **top**
  - a logical, if true the caption is placed on top
- **outfile**
  - the path to the file
- **clipboard**
  - if clipboard TRUE, a copy to the clipboard is made
- **append**
  - is the file appended to the previous one ?
- **digits**
  - the number of digits
- **...**
  - additional parameters to be passed to the function
Value
an xtable

Author(s)
Cedric Briand <cedric.briand@eptb-vilaine.fr>

---

**induk**
*unique values of a vector*

---

**Description**
returns the index of values appearing only once in a vector: match(unique(a),a), replicated values
are not returned on their second occurrence

**Usage**

```r
induk(a)
```

**Arguments**
a a vector

**Value**
the index unique values within a vector

**Author(s)**
Cedric Briand <cedric.briand@eptb-vilaine.fr>

---

**is.even**
is.even function modified from package sma

---

**Description**
is.even function modified from package sma (which did not verify that the entry was indeed an
integer)

**Usage**

```r
is.even(x)
```

**Arguments**
x integer
is.odd

Description
id.odd function modified from package sma (which did not verify that the entry was indeed an integer)

Usage
is.odd(x)

Arguments
x integer

Value
a logical

Author(s)
Adapted from Henrik Bengtsson

killfactor

Description
very useful function remove factor that appear, noticeably after loading with ODBC

Usage
killfactor(df)

Arguments
df a data.frame
RequeteODBC-class

Value

```
  df
```

Author(s)

Cedric Briand <cedric.briand"at"eptb-vilaine.fr>

<table>
<thead>
<tr>
<th>RequeteODBC-class</th>
<th>Class &quot;RequeteODBC&quot;</th>
</tr>
</thead>
</table>

Description

ODBC Query. This class enables to retrieve data from the database. This class is inherited by RequeteODBCwhere and RequeteODBCwheredate

Objects from the Class

Objects can be created by calls of the form new("RequeteODBC", sql=character(), query=data.frame()).

- baseODBC: Object of class "vector" The name, user and password of the database
- connection: Object of class "ANY" The connection
- etat: Object of class "character" The state of the query (Connecting, successful,...)
- silent: Object of class "logical" True if the query must be executed silently, FALSE
- sql: Object of class "character" The query
- query: Object of class "data.frame" The result of the query
- open: Object of class "logical" Should the connection remain open, choosing this ensures more rapid multiple queries

Extends

Class "ConnectionODBC", directly.

Methods

- `connect` signature(objet = "RequeteODBC"): Connection to the database

Note

Inherits from ConnectionODBC

Author(s)

cedric.briand"at"eptb-vilaine.fr

See Also

ConnectionODBC RequeteODBCwhere RequeteODBCwheredate
**Examples**

```r
showClass("RequeteODBC")
## Not run:
objet=new("RequeteODBC")
objet@open=TRUE  # this will leave the connection open, by default it closes after the query is sent
# the following will work only if you have configured and ODBC link
objet@baseODBC="myODBCconnection","myusername","mypassword"
objet@sql= "select * from mytable limit 100"
objet<connect(objet)
odbcclose(objet@connection)
envir_stacomi=newNenv()
C while testing I like to see the output of sometimes complex queries generated by the program
assign("showmerequest",1,envir_stacomi)  # can be anything just tests the existence of "showmerequest" in envir_stacomi
objet=new("RequeteODBC")
objet@baseODBC="myODBCconnection","myusername","mypassword"
objet@sql= "select * from mytable limit 100"
objet<connect(objet)
# the connection is already closed, the query is printed
## End(Not run)
```

---

**RequeteODBCwhere-class**

*Class "RequeteODBCwhere"*

---

**Description**

SQL Query with WHERE and ORDER BY clauses.

**Objects from the Class**

Objects can be created by calls of the form `new("RequeteODBCwhere", where=character(), and=vector(), order_by=character(), sql=character())`.

- **select**: Object of class "character" The "SELECT" part of the query
- **where**: Object of class "character" The "WHERE" part of the query
- **and**: Object of class "vector" The "AND" part of the query
- **order_by**: Object of class "character" The "ORDER BY" part of the query
- **sql**: Object of class "character" The query built by aggregating "select","where","and", and "order_by" slots
- **query**: Object of class "data.frame" The result of the query
- **open**: Object of class "logical" Should the connection remain open, choosing this ensures more rapid multiple queries
- **baseODBC**: Object of class "vector" The name, user and password of the database
- **silent**: Object of class "logical" TRUE if the query must be executed silently, FALSE else
- **etat**: Object of class "character" The state of the query (Connecting, successful,...)
- **connection**: Object of class "ANY" The database connection
**Extends**

Class "RequeteODBC", directly. Class "ConnectionODBC", by class "RequeteODBC", distance 2.

**Methods**

```r
connect signature(objet = "RequeteODBCwhere"): Connect to the database
```

**Note**

Inherits from RequeteODBC the syntax is where="WHERE ..." and =vector("AND...","AND...") order_by="ORDER BY .." The query will syntax will be printed upon failure.

**Author(s)**

cedric.briand"at"eptb-vilaine.fr

**See Also**

ConnectionODBC RequeteODBC RequeteODBCwheredate

**Examples**

```r
showClass("RequeteODBCwhere")
## Not run:

test<-

objet=new("RequeteODBCwhere")
objet@baseODBC=c("myodbcconnection","myusername","mypassword")
objet@select= "select * from mytable limit 100"
# assuming mycol, mycol1 and mycol2 are numeric
objet@where=paste(" where mycol","test","sep="")
objet@and=paste(" and mycol2","test"," and mycol3<","test","sep="")
objet@order_by=" order by mycol1"
objet<-connect(objet)
# now objet@sql contains the syntax of the query. By changing the test variable, one can see how the
# function might be usefull
# objet@query contains the resulting data.frame

## End(Not run)
```
Objects from the Class

Objects can be created by calls of the form `new("RequeteODBCwheredate", datedebut="POSIXlt", datefin="POSIXlt",
datedebut: Object of class "POSIXlt" ~ The starting date
datefin: Object of class "POSIXlt" ~ The ending date
colonnebegin: Object of class "character" ~ The name begin column
colonnefin: Object of class "character" ~ The name end column

Slots

datedebut: Object of class "POSIXlt" ~ The starting date
datefin: Object of class "POSIXlt" ~ The ending date
colonnebegin: Object of class "character" ~ The name begin column
colonnefin: Object of class "character" ~ The name end column
where: Object of class "character" ~ The WHERE clause
and: Object of class "vector" ~ The AND clause
order_by: Object of class "character" ~ The ORDER BY clause
sql: Object of class "character" ~ The SELECT clause
query: Object of class "data.frame" ~ The result of the query
baseODBC: Object of class "vector" ~ The database
silent: Object of class "logical" ~ The mode
etat: Object of class "character" ~ The state
connection: Object of class "ANY" ~ The connection

Extends

Class "RequeteODBCwhere", directly. Class "RequeteODBC", by class "RequeteODBCwhere", distance 2. Class "ConnectionODBC", by class "RequeteODBCwhere", distance 3.

Methods

connect signature(objet = "RequeteODBCwheredate"): Connexion to the database

Note

Inherits from RequeteODBCwhere and uses its connect method with a new SetAs. This function is only useful in databases supporting the "overlaps" statement.

Author(s)

cedric.briand"at"eptb-vilaine.fr

See Also

ConnectionODBC RequeteODBC RequeteODBCwhere
Examples

showClass("RequeteODBCwhere date")

---

**tab2df**  
*Function to transform a ftable into dataframe but just keeping the counts, works with ftable of dim 2*

---

**Description**

Function to transform a ftable into dataframe but just keeping the counts works with ftable of dim 2

**Usage**

`tab2df(tab)`

**Arguments**

- `tab`  
a flat table

**Author(s)**

Cedric Briand <cedric.briand"at"eptb-vilaine.fr>
Index

*Topic **classes**
  - ConnectionODBC-class, 4
  - RequeteODBC-class, 8
  - RequeteODBCwhere-class, 9

*Topic **methods**
  - connect-methods, 3

*Topic **package**
  - stacomirtools-package, 2

chnames, 2
connect (connect-methods), 3
connect, ConnectionODBC-method (connect-methods), 3
connect, RequeteODBC-method (connect-methods), 3
connect, RequeteODBCwhere-method (connect-methods), 3
connect, RequeteODBCwheredate-method (connect-methods), 3
connect-methods, 3
ConnectionODBC, 8, 10, 11
ConnectionODBC (ConnectionODBC-class), 4
ConnectionODBC-class, 4
ex, 5
funhtml, 5
induk, 6
is.even, 6
is.odd, 7
killfactor, 7
RequeteODBC, 10, 11
RequeteODBC (RequeteODBC-class), 8
RequeteODBC-class, 8
RequeteODBCwhere, 8, 11
RequeteODBCwhere-class, 9
RequeteODBCwheredate, 8, 10
RequeteODBCwheredate-class, 10

stacomirtools-package, 2
tab2df, 12