

Rdbi

November 11, 2009

R topics documented:

dbAppendTable	1
dbClearResult	2
dbColumnInfo	3
dbConnectionInfo	3
dbConnect	4
dbDisconnect	5
dbGetQuery	6
dbGetResult	7
dbListTables	8
dbReadTable	8
dbReconnect	9
dbResultInfo	10
dbSendQuery	11
dbWriteTable	12
Default methods and internal functions	12
Utility functions	14

Index	16
--------------	-----------

dbAppendTable	<i>Appends data to a database table</i>
---------------	---

Description

dbAppendTable is a generic function that, when called on a valid database connection object, appends the contents of a data frame to a database table.

Usage

```
dbAppendTable(conn, ...)
```

Arguments

conn	A database connection object.
...	Additional arguments

Details

Column names of the data frame must match column names of the database table. Implementations should allow the data frame columns to be a subset of the database table columns and match R column names to SQL column names. Appends must be atomic. Implementations must use transactions or emulate them.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [methods](#), [class](#), [on.exit](#)

dbClearResult

Clears resources associated with a query result

Description

dbClearResult is a generic function that, when called on a result object, clears any resources associated with that object.

Usage

```
dbClearResult(result)
```

Arguments

result A query result object.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbSendQuery](#), [data.frame](#)

dbColumnInfo	<i>Returns type information about a result column</i>
--------------	---

Description

dbColumnInfo is a generic function that, when called on a result object, returns type information about tuple fields in the result.

Usage

```
dbColumnInfo(result)
```

Arguments

result A query result object.

Value

Returns a data frame with each row corresponding to a different field in the result. Rows are named for each field. Any available information about a field can be presented in columns of the data frame. The most important of these is probably a "Type" column that can be used to convert strings returned by the query into appropriate R types.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbSendQuery](#), [dbResultInfo](#)

dbConnectionInfo	<i>Returns a list of connection status attributes</i>
------------------	---

Description

dbConnectionInfo is a generic function that, when called on a valid connection object, returns a list containing connection status information. It is called by `print.Rdbi.conn`.

Usage

```
dbConnectionInfo(conn)
```

Arguments

conn A database connection object.

Details

Any useful information such as the database host and connection status should be returned.

Value

An arbitrary list of connection attributes.

Note

I should probably define a `dbConnectionOK` method that is generic and return `TRUE` when the connection is valid. However, you don't need to constantly check for a valid connection object. Keep the code path short! For example, in `Rdbi.PgSQL`, there is a C function that submits a query request to the database backend. This is the only time that the connection object is actually dereferenced to its connection pointer. This C function checks for a valid connection and returns and error if needed. Therefore there is no reason to check whether the connection is valid before passing it to a query function; the C code will do the check. In this way the interface is simplified and the connection checking is localized to a single call instead of scattered all over the code. As Bertrand Meyer put it: "*Defensive coding is offensive!*".

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [dbDisconnect](#), [methods](#), [class](#)

dbConnect

Connect to a database

Description

`dbConnect` is a generic function that, when evoked with a valid database class, will return a connection object.

Usage

```
dbConnect(dbObj, ...)
```

Arguments

`dbObj` A database class object.

`...` An argument list specifying connection options.

Details

Each package that sub-classes `Rdbi` must provide a `dbConnect` function. The first argument of the `dbConnect` function is an object whose class determines which `dbConnect` method is actually called. For example, the `Rdbi.PgSQL` package provides a function `PgSQL()` that returns an object of class `c("PgSQL", "Rdbi")`. Therefore the call `dbConnect(PgSQL(), ...)` will invoke the method `dbConnect.PgSQL`. `Rdbi` arranges for the specific package to be loaded via the `autoload` mechanism. In this example, `Rdbi.PgSQL` is autoloaded when `PgSQL` is called.

Value

A database connection object that inherits from `"Rdbi.conn"`. Additionally, the connection object should possess two attributes required to reopen the connection from the object. The `"library.call"` contains a call or expression that will load the library necessary to support the connection object. The `"connect.call"` attribute should contain the call that created the connection object. Also, it is very convenient to arrange for low-level connection resources to be freed when the R connection object is garbage collected. See the `Rdbi.PgSQL C` code for an example.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbDisconnect](#), [dbReconnect](#), [methods](#), [class](#), [match.call](#)

dbDisconnect

Closes a database connection

Description

`dbDisconnect` a generic function that, when called with a valid connection object, closes the database connection and frees any resource associated with the connection.

Usage

```
dbDisconnect(conn)
```

Arguments

`conn` A database connection object.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [dbReconnect](#), [methods](#), [class](#)

dbGetQuery

Submit a query string and fetch results

Description

`dbGetQuery` is a generic function that, when called on a valid connection object, executes a query and returns a dataframe with the query results, or an error if no results were generated.

Usage

```
dbGetQuery(conn, ...)
```

Arguments

<code>conn</code>	A database connection object
<code>...</code>	Arguments that when pasted together form a query string

Details

Simply calls [dbSendQuery](#) and [dbGetResult](#).

Value

A dataframe with the results.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [dbGetResult](#), [methods](#), [class](#), [paste](#)

dbGetResult	<i>Fetch results from a query</i>
-------------	-----------------------------------

Description

dbGetResult is a generic function that, when called on a result object, returns any tuples associated with the object.

Usage

```
dbGetResult(result, as.matrix = FALSE)
```

Arguments

result A query result object.

as.matrix A boolean to indicate whether the results will be returned as a matrix

Details

Fetches the results of a query and returns a dataframe. Non-character types should probably be converted to the appropriate numeric or logical type. A generic type conversion interface is still needed.

Value

A dataframe with the results.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbSendQuery](#), [data.frame](#)

dbListTables *Lists database tables*

Description

A generic function that, when called on a valid connection object, returns a list of tables stored in the database backend.

Usage

```
dbListTables(conn, ...)
```

Arguments

conn A database connection object.
... Additional arguments.

Value

A list of character strings with table names.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [ls](#), [methods](#), [class](#), [match.call](#)

dbReadTable *Reads a table into a data frame*

Description

dbReadTable is a generic function that, when called on a valid connection object, reads a table from the database backend and returns a data frame with the contents.

Usage

```
dbReadTable(conn, ...)
```

Arguments

conn A database connection object.
... Additional arguments.

Details

SQL types should be cast to R types to the extent possible. A generic mechanism for type conversion is lacking.

Value

A data frame.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [methods](#), [class](#)

dbReconnect

Reopens a connection to a database

Description

`dbReconnect` a generic function that, when called with a valid connection object, reopens the connection to the database backend.

Usage

```
dbReconnect(conn)
```

Arguments

`conn` A database connection object.

Details

A database connection object contains the necessary information to re-establish a connection. Thus, a database connection object can be saved across R sessions and reconnected later. If I can convince the R developers to add a generic function that is always called when objects are restored, then it will be possible to have connections persist across R sessions.

Note that packages that implement the Rdbi interface do not need to provide a `dbReconnect` function as long as the connection object returned by `dbConnect` inherits from "Rdbi.conn" and has attributes described in the documentation for `dbConnect`. `dbReconnect.Rdbi.conn` can reconnect the object. If `dbReconnect.Rdbi.conn` is not general enough, the package can provide its own method.

Value

A database connection object.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [dbDisconnect](#), [methods](#), [class](#)

dbResultInfo

Returns information about a query result

Description

dbResultInfo is a generic function that, when called on a result object, returns a list with status information.

Usage

```
dbResultInfo(result)
```

Arguments

result A query result object.

Value

A list with result status information.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbSendQuery](#)

dbSendQuery	<i>Submits a query string to the database backend</i>
-------------	---

Description

dbSendQuery is a generic function that, when called on a valid connection object, pastes its arguments into a query string and submits it to the database backend for processing.

Usage

```
dbSendQuery(conn, ...)
```

Arguments

conn	A database connection object.
...	Arguments that when pasted together form a query string.

Details

Sub-classed dbSendQuery methods should not fail unless the connection is not valid, in which case an error message should be printed. Information about the query result status can be obtained by dereferencing the returned result object.

Value

A result object inheriting from "Rdbi.result". You can arrange for the result buffer to be cleared when the result object is garbage collected by registering a finalizer function. See the C code in Rdbi.PgSQL.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [dbGetResult](#), [dbResultInfo](#), [methods](#), [class](#), [paste](#)

<code>dbWriteTable</code>	<i>Writes a data frame to a database table</i>
---------------------------	--

Description

`dbWriteTable` is a generic function that, when called on a valid connection object, write a data frame to a database table.

Usage

```
dbWriteTable(conn, ...)
```

Arguments

<code>conn</code>	A database connection object.
<code>...</code>	Additional arguments.

Details

Any writes to the database backend should be atomic. Packages subclassing `Rdbi` need to use or emulate transactions.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

See Also

[dbConnect](#), [methods](#), [class](#), [on.exit](#)

Default methods and internal functions
Default methods and internal functions

Description

Default methods and internal functions

Usage

```
## Default S3 method:
dbAppendTable(conn, ...)
## Default S3 method:
dbClearResult(result)
## Default S3 method:
dbColumnInfo(result)
## Default S3 method:
dbConnect(dbObj, ...)
## Default S3 method:
dbConnectionInfo(conn)
## Default S3 method:
dbDisconnect(conn)
## Default S3 method:
dbGetQuery(conn, ...)
## Default S3 method:
dbGetResult(result, as.matrix)
## Default S3 method:
dbListTables(conn, ...)
## Default S3 method:
dbReadTable(conn, ...)
## Default S3 method:
dbReconnect(conn)
## Default S3 method:
dbResultInfo(result)
## Default S3 method:
dbSendQuery(conn, ...)
## Default S3 method:
dbWriteTable(conn, ...)
## S3 method for class 'Rdbi.conn':
print(x, ...)
## S3 method for class 'Rdbi.result':
print(x, ...)
```

Note

See source for details.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

Description

`list.to.csv` converts a list of values to a comma separated string.

`list.to.key.pair.string` takes a list and reformats it as a string with arbitrary separators between keys and values and between key-value pairs.

`printListPairs` is a simple utility function that prints list key-value pairs nicely formatted.

`single.quote` pastes its arguments into a single quoted string.

`double.quote` pastes its arguments into a double quoted string.

`strip.line.feeds` removes line feeds from a string.

`expand.asis` takes an "asis" argument as used in `read.table` and returns a vector of booleans indicating whether columns should be converted to factors.

Usage

```
list.to.csv(...)  
list.to.key.pair.string(key.list, key.sep = "=", pair.sep = " ")  
printListPairs(list)  
single.quote(...)  
double.quote(...)  
strip.line.feeds(x)  
expand.asis(as.is, len = length(as.is))
```

Arguments

<code>list</code>	A list of values.
<code>key.sep</code>	Separator between keys and values.
<code>pair.sep</code>	Separator between key-value pairs.
<code>string</code>	A text string.
<code>as.is</code>	A column name or number, or a vector of column numbers, or a vector of booleans indicating which columns to leave in character mode as opposed to converting to factors.
<code>len</code>	The length of the output "asis" vector.

Author(s)

Timothy H. Keitt

References

<http://rdbi.sourceforge.net/>

Examples

```
list.to.csv(list(a = 'a', b = 'b', c = 'c'))
printListPairs(list(a = 'a', b = 'b', c = 'c'))
list.to.key.pair.string(list(a = 'a', b = 'b', c = 'c'),
  key.sep = " -> ", pair.sep = " | ")
single.quote("test")
double.quote("test")
expand.asis(5, 10)
strip.line.feeds("test\n")
```

Index

*Topic **connection**

- dbAppendTable, 1
- dbConnect, 4
- dbConnectionInfo, 3
- dbDisconnect, 5
- dbGetQuery, 5
- dbListTables, 7
- dbReadTable, 7
- dbReconnect, 8
- dbSendQuery, 10
- dbWriteTable, 11

*Topic **data**

- dbAppendTable, 1
- dbClearResult, 2
- dbConnect, 4
- dbConnectionInfo, 3
- dbDisconnect, 5
- dbGetQuery, 5
- dbGetResult, 6
- dbListTables, 7
- dbReadTable, 7
- dbReconnect, 8
- dbSendQuery, 10
- dbWriteTable, 11

*Topic **internal**

- Default methods and internal functions, 11
- Utility functions, 13

*Topic **methods**

- dbAppendTable, 1
- dbClearResult, 2
- dbColumnInfo, 2
- dbConnect, 4
- dbConnectionInfo, 3
- dbDisconnect, 5
- dbGetQuery, 5
- dbGetResult, 6
- dbListTables, 7
- dbReadTable, 7
- dbReconnect, 8
- dbResultInfo, 9
- dbSendQuery, 10
- dbWriteTable, 11

autoload, 4

class, 1, 4–11

data.frame, 2, 6

dbAppendTable, 1

dbAppendTable.default (*Default methods and internal functions*), 11

dbClearResult, 2

dbClearResult.default (*Default methods and internal functions*), 11

dbColumnInfo, 2

dbColumnInfo.default (*Default methods and internal functions*), 11

dbConnect, 1, 4, 4–11

dbConnect.default (*Default methods and internal functions*), 11

dbConnectionInfo, 3

dbConnectionInfo.default (*Default methods and internal functions*), 11

dbDisconnect, 4, 5, 9

dbDisconnect.default (*Default methods and internal functions*), 11

dbGetQuery, 5

dbGetQuery.default (*Default methods and internal functions*), 11

dbGetResult, 5, 6, 6, 10

dbGetResult.default (*Default methods and internal functions*), 11

dbListTables, 7

dbListTables.default (*Default methods and internal functions*), 11

dbReadTable, 7

dbReadTable.default (*Default methods and internal*

functions), 11

dbReconnect, 4, 5, 8

dbReconnect.default (Default methods and internal functions), 11

dbReconnect.Rdbi.conn (dbReconnect), 8

dbResultInfo, 3, 9, 10

dbResultInfo.default (Default methods and internal functions), 11

dbSendQuery, 2, 3, 5, 6, 9, 10

dbSendQuery.default (Default methods and internal functions), 11

dbWriteTable, 11

dbWriteTable.default (Default methods and internal functions), 11

Default methods and internal functions, 11

double.quote (Utility functions), 13

expand.asis (Utility functions), 13

list.to.csv (Utility functions), 13

list.to.key.pair.string (Utility functions), 13

ls, 7

match.call, 4, 7

methods, 1, 4–11

on.exit, 1, 11

paste, 6, 10

print.Rdbi.conn (Default methods and internal functions), 11

print.Rdbi.result (Default methods and internal functions), 11

printListPairs (Utility functions), 13

read.table, 13

single.quote (Utility functions), 13

strip.line.feeds (Utility functions), 13

Utility functions, 13