

# Package ‘RODBCDBI’

August 29, 2016

**Type** Package

**Version** 0.1.1

**Title** Provides Access to Databases Through the ODBC Interface

**Description** An implementation of R's DBI interface using ODBC package as a back-end. This allows R to connect to any DBMS that has a ODBC driver.

**License** MIT + file LICENSE

**Imports** methods, DBI, RODBC

**Suggests** testthat

**Collate** 'RODBCDBI.R' 'ODBCConnection.R' 'ODBCDriver.R' 'ODBCResult.R'

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Nagi Teramo [aut, cre],  
Shinichi Takayanagi [aut]

**Maintainer** Nagi Teramo <teramonagi@gmail.com>

**Repository** CRAN

**Date/Publication** 2016-03-14 07:54:09

## R topics documented:

dbConnect,ODBCDriver-method . . . . .	2
dbDisconnect,ODBCConnection-method . . . . .	2
dbExistsTable,ODBCConnection,character-method . . . . .	3
dbGetInfo,ODBCConnection-method . . . . .	3
dbListFields,ODBCConnection,character-method . . . . .	4
dbListTables,ODBCConnection-method . . . . .	4
dbReadTable,ODBCConnection,character-method . . . . .	5
dbRemoveTable,ODBCConnection,character-method . . . . .	6
dbSendQuery,ODBCConnection-method . . . . .	6
dbWriteTable,ODBCConnection,character,data.frame-method . . . . .	7
ODBC . . . . .	8
odbc-meta . . . . .	8
RODBCDBI . . . . .	9

**Index****10**


---

 dbConnect,ODBCDriver-method

*Connect/disconnect to a ODBC data source*


---

**Description**

These methods are straight-forward implementations of the corresponding generic functions.

**Usage**

```
## S4 method for signature 'ODBCDriver'
dbConnect(drv, dsn, user = NULL, password = NULL,
  ...)
```

**Arguments**

drv	an object of class ODBCDriver
dsn	Data source name you defined by ODBC data source administrator tool.
user	User name to connect as.
password	Password to be used if the DSN demands password authentication.
...	Other parameters passed on to methods

**Examples**

```
## Not run:
# Connect to a ODBC data source
con <- dbConnect(RODBCDBI::ODBC(), dsn="test")
# Always cleanup by disconnecting the database
#' dbDisconnect(con)

## End(Not run)
```

---

 dbDisconnect,ODBCConnection-method

*Close a current session.*


---

**Description**

Close a current session.

**Usage**

```
## S4 method for signature 'ODBCConnection'
dbDisconnect(conn)
```

**Arguments**

conn            a [ODBCConnection](#) object, produced by [dbConnect](#)

**Examples**

```
## Not run:
library(DBI)
con <- dbConnect(RODBCDBI::ODBC(), dsn="test", user="sa", password="Password12!")
dbDisconnect(con)

## End(Not run)
```

---

dbExistsTable, ODBCConnection, character-method

*Does the table exist?*

---

**Description**

Does the table exist?

**Usage**

```
## S4 method for signature 'ODBCConnection, character'
dbExistsTable(conn, name)
```

**Arguments**

conn            An existing [ODBCConnection](#)

name            String, name of table. Match is case insensitive.

**Value**

boolean value which indicated whether the table exists or not

---

dbGetInfo, ODBCConnection-method

*Get DBMS metadata.*

---

**Description**

Get DBMS metadata.

**Usage**

```
## S4 method for signature 'ODBCConnection'
dbGetInfo(dbObj, ...)
```

**Arguments**

dbObj            An object inheriting from [ODBCConnection](#), [ODBCDriver](#), or a [ODBCResult](#)  
 ...             Other parameters passed on to methods

---

dbListFields,ODBCConnection,character-method  
*List fields in specified table.*

---

**Description**

List fields in specified table.

**Usage**

```
## S4 method for signature 'ODBCConnection,character'
dbListFields(conn, name)
```

**Arguments**

conn            An existing [ODBCConnection](#)  
 name            a length 1 character vector giving the name of a table.

**Examples**

```
## Not run:
library(DBI)
con <- dbConnect(RODBCDBI::ODBC(), dsn="test", user="sa", password="Password12!")
dbWriteTable(con, "iris", iris, overwrite=TRUE)
dbListFields(con, "iris")
dbDisconnect(con)

## End(Not run)
```

---

dbListTables,ODBCConnection-method  
*List available ODBC tables.*

---

**Description**

List available ODBC tables.

**Usage**

```
## S4 method for signature 'ODBCConnection'
dbListTables(conn)
```

**Arguments**

conn                    An existing [ODBCConnection](#)

---

dbReadTable,ODBCConnection,character-method

*Convenience functions for importing/exporting DBMS tables*

---

**Description**

These functions mimic their R/S-Plus counterpart `get`, `assign`, `exists`, `remove`, and `objects`, except that they generate code that gets remotely executed in a database engine.

**Usage**

```
## S4 method for signature 'ODBCConnection,character'
dbReadTable(conn, name, row.names = NA,
             check.names = TRUE, select.cols = "*")
```

**Arguments**

conn                    a [ODBCConnection](#) object, produced by [dbConnect](#)

name                    a character string specifying a table name.

row.names              a character string specifying a table name.

check.names            If TRUE, the default, column names will be converted to valid R identifiers.

select.cols            A SQL statement (in the form of a character vector of length 1) giving the columns to select. E.g. "\*" selects all columns, "x,y,z" selects three columns named as listed.

**Value**

A data.frame in the case of `dbReadTable`; otherwise a logical indicating whether the operation was successful.

**Note**

Note that the data.frame returned by `dbReadTable` only has primitive data, e.g., it does not coerce character data to factors.

**Examples**

```
## Not run:
library(DBI)
con <- dbConnect(RODBCDBI::ODBC(), dsn="test", user="sa", password="Password12!")
dbWriteTable(con, "mtcars", mtcars, overwrite=TRUE)
dbReadTable(con, "mtcars")
dbGetQuery(con, "SELECT * FROM mtcars WHERE cyl = 8")
```

```
# Suppress row names
dbReadTable(con, "mtcars", row.names = FALSE)
dbGetQuery(con, "SELECT * FROM mtcars WHERE cyl = 8", row.names = FALSE)

dbDisconnect(con)

## End(Not run)
```

---

dbRemoveTable,ODBCConnection,character-method  
*Remove a table from the database.*

---

### Description

Executes the SQL DROP TABLE.

### Usage

```
## S4 method for signature 'ODBCConnection,character'
dbRemoveTable(conn, name)
```

### Arguments

conn	An existing <a href="#">ODBCConnection</a>
name	character vector of length 1 giving name of table to remove

---

dbSendQuery,ODBCConnection-method  
*Execute a statement on a given database connection.*

---

### Description

To retrieve results a chunk at a time, use dbSendQuery, dbFetch, then ClearResult. Alternatively, if you want all the results (and they'll fit in memory) use dbGetQuery which sends, fetches and clears for you.

### Usage

```
## S4 method for signature 'ODBCConnection'
dbSendQuery(conn, statement, ...)

## S4 method for signature 'ODBCResult'
dbFetch(res, n = -1, ...)

## S4 method for signature 'ODBCResult'
dbHasCompleted(res, ...)
```

```
## S4 method for signature 'ODBCResult'
dbClearResult(res, ...)
```

### Arguments

conn	An existing <a href="#">ODBCConnection</a>
statement	The SQL which you want to run
...	Other parameters passed on to methods
res	An object of class <a href="#">ODBCResult</a>
n	Number of rows to return. If less than zero returns all rows.

---

```
dbWriteTable,ODBCConnection,character,data.frame-method
Write a local data frame or file to the database.
```

---

### Description

Write a local data frame or file to the database.

### Usage

```
## S4 method for signature 'ODBCConnection,character,data.frame'
dbWriteTable(conn, name, value,
  overwrite = FALSE, append = FALSE, ...)
```

### Arguments

conn	a <a href="#">ODBCConnection</a> object, produced by <a href="#">dbConnect</a>
name	a character string specifying a table name. <a href="#">ODBCConnection</a> table names are <i>not</i> case sensitive, e.g., table names ABC and abc are considered equal.
value	a data.frame (or coercible to data.frame) object or a file name (character). when value is a character, it is interpreted as a file name and its contents imported to ODBC.
overwrite	logical. Should data be overwritten?
append	logical. Should data be appended to an existing table?
...	additional arguments passed to the generic.

### Examples

```
## Not run:
library(DBI)
con <- dbConnect(RODBCDBI::ODBC(), dsn="test", user="sa", password="Password12!")
dbWriteTable(con, "mtcars", mtcars, overwrite=TRUE)
dbReadTable(con, "mtcars")
dbDisconnect(con)

## End(Not run)
```

---

ODBC

*Generate an object of ODBCdriver class*

---

### Description

This driver is for implementing the R database (DBI) API. This class should always be initialized with the ODBC() function. ODBC driver does nothing for ODBC connection. It just exists for S4 class compatibility with DBI package.

### Usage

```
ODBC()
```

### Examples

```
## Not run:
driver <- RODBCDBI::ODBC()
# Connect to a ODBC data source
con <- dbConnect(driver, dsn="test")
# Always cleanup by disconnecting the database
#' dbDisconnect(con)

## End(Not run)
```

---

odbc-meta

*Database interface meta-data.*

---

### Description

See documentation of generics for more details.

### Usage

```
## S4 method for signature 'ODBCResult'
dbGetRowCount(res, ...)

## S4 method for signature 'ODBCResult'
dbGetStatement(res, ...)

## S4 method for signature 'ODBCResult'
dbGetInfo(dbObj, ...)

## S4 method for signature 'ODBCResult'
dbColumnInfo(res, ...)
```



**Arguments**

res            An object of class `ODBCResult`  
 ...           Ignored. Needed for compatibility with generic  
 dbObj        An object inheriting from `ODBCConnection`, `ODBCDriver`, or a `ODBCResult`

**Examples**

```
## Not run:
library(DBI)
data(USArrests)
con <- dbConnect(RODBCDBI::ODBC(), dsn="test", user="sa", password="Password12!")
dbWriteTable(con, "t1", USArrests, overwrite=TRUE)
dbWriteTable(con, "t2", USArrests, overwrite=TRUE)

dbListTables(con)

rs <- dbSendQuery(con, "select * from t1 where UrbanPop >= 80")
dbGetStatement(rs)
dbHasCompleted(rs)

info <- dbGetInfo(rs)
names(info)
info$fields

dbFetch(rs, n=2)
dbHasCompleted(rs)
info <- dbGetInfo(rs)
info$fields
dbClearResult(rs)

# DBIConnection info
names(dbGetInfo(con))

dbDisconnect(con)

## End(Not run)
```

---

RODBCDBI

*RODBCDBI*


---

**Description**

Provides Access to Databases Through the ODBC Interface An implementation of R's DBI interface using ODBC package as a back-end. This allows R to connect to any DBMS that has a ODBC driver.

# Index

dbClearResult, ODBCResult-method  
(dbSendQuery, ODBCConnection-method), RODBCDBI, 9  
6

dbColumnInfo, ODBCResult-method  
(odbc-meta), 8

dbConnect, 3, 5, 7

dbConnect, ODBCDriver-method, 2

dbDisconnect, ODBCConnection-method, 2

dbExistsTable, ODBCConnection, character-method,  
3

dbFetch, ODBCResult-method  
(dbSendQuery, ODBCConnection-method),  
6

dbGetInfo, ODBCConnection-method, 3

dbGetInfo, ODBCResult-method  
(odbc-meta), 8

dbGetRowCount, ODBCResult-method  
(odbc-meta), 8

dbGetStatement, ODBCResult-method  
(odbc-meta), 8

dbHasCompleted, ODBCResult-method  
(dbSendQuery, ODBCConnection-method),  
6

dbListFields, ODBCConnection, character-method,  
4

dbListTables, ODBCConnection-method, 4

dbReadTable, ODBCConnection, character-method,  
5

dbRemoveTable, ODBCConnection, character-method,  
6

dbSendQuery, ODBCConnection-method, 6

dbWriteTable, ODBCConnection, character, data.frame-method,  
7

ODBC, 8

odbc-meta, 8

ODBCConnection, 3-7, 9

ODBCDriver, 4, 9

ODBCResult, 4, 7, 9