Rdbi

October 24, 2011

`dbAppendTable` *Appends data to a database table*

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` *Returns type information about a result column*

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`

`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`

`dbConnectionInfo` *Returns a list of connection status attributes*

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 derefenced 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`

`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

`conn`	A database connection object
`...`	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` *Fetch results from a query*

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 *Submits a query string to the database backend*

Description

dbSendQuery is a generic function that, when called on a valid connection object, pastes is argments 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

`dbWriteTable` *Writes a data frame to a database table*

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

`conn`	A database connection object.
`...`	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`

Index