

Package ‘bcputility’

February 20, 2022

Type Package

Title Wrapper for SQL Server bcp Utility

Version 0.3.0

Description Provides functions to utilize a command line utility that does bulk inserts and exports from SQL Server databases.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

SystemRequirements bcp Utility

Imports data.table, DBI, odbc, sf

URL <https://bcputility.roh.engineering>,
<https://github.com/tomroh/bcputility>

BugReports <https://github.com/tomroh/bcputility/issues>

NeedsCompilation no

Author Thomas Roh [aut, cre]

Maintainer Thomas Roh <thomas@roh.engineering>

Repository CRAN

Date/Publication 2022-02-20 22:00:02 UTC

R topics documented:

bcpExport	2
bcpImport	3
Index	6

bcpExport

Export data from SQL Server

Description

A wrapper for a system call to the bcp utility which writes a SQL Server table or query (T-SQL) to a file.

Usage

```
bcpExport(
  file,
  server,
  database,
  table,
  query,
  trustedconnection = TRUE,
  username,
  password,
  fieldterminator = "\t",
  rowterminator = ifelse(.Platform$OS.type == "windows", "\r\n", "\n"),
  datatypes = c("char", "nchar"),
  ...
)
```

Arguments

file	output file name
server	the instance of SQL Server to which to connect
database	Specifies the database to connect to
table	name of the source table when exporting from SQL Server
query	Transact-SQL query that returns a result set. Ignored if table is specified.
trustedconnection	use integrated security, username and password are not required
username	login ID
password	password for login ID
fieldterminator	character separator for columns
rowterminator	character separator for rows–new lines
datatypes	the format of datatypes, char performs the operation using a character data type, nchar performs the bulk copy operation using Unicode characters
...	arguments to pass system2

Value

No return value. Operations from bcp are printed to console; see . . . to redirect output

Examples

```
## Not run:
bcpExport('myfile.tsv',
          server = server,
          database = database,
          table = 'mytable',
          fieldterminator = '|',
          stdout = FALSE,
          datatypes = 'char')

## End(Not run)
```

bcpImport

Import data to SQL Server

Description

A wrapper for a system call to the bcp utility which bulk inserts to SQL Server.

Usage

```
bcpImport(
  x,
  server,
  database,
  table,
  driver = "SQL Server",
  maxerrors = 10,
  batchsize = 1000,
  packetsize = 4096,
  regional = FALSE,
  trustedconnection = TRUE,
  username,
  password,
  fieldterminator = "\t",
  rowterminator = ifelse(.Platform$OS.type == "windows", "\r\n", "\n"),
  overwrite = FALSE,
  spatialtype = c("geometry", "geography"),
  ...
)
```

Arguments

x	dataframe object or path to file
server	the instance of SQL Server to which to connect
database	specifies the database to connect to
table	Name of the source table when exporting from SQL Server. For specifying the schema in the table name see <code>DBI::SQL</code> or <code>DBI::Id</code> .
driver	name of driver for ODBC connection
maxerrors	maximum number of errors allowed
batchsize	number of rows to write at a time; 10,000 to 50,000 is a starting recommendation
packetsize	size of packets to be sent
regional	Specifies that currency, date, and time data is bulk copied into SQL Server using the regional format defined for the locale setting of the client computer
trustedconnection	use integrated security, username and password are not required
username	login ID
password	password for login ID
fieldterminator	character separator for columns
rowterminator	character separator for rows—new lines
overwrite	Whether to overwrite the table if it exists
spatialtype	spatial data type for schema https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-types-overview , ignored if x is not an 'sf' object
...	arguments to pass to <code>system2</code>

Details

If x is a dataframe object, `data.table::fwrite` is used to write the in memory object to disk in a temporary file that is deleted when the function exits. The `fieldterminator` and `rowterminator` are ignored in this case.

If `overwrite` is TRUE, any existing table of the same name will be deleted and the schema is inferred from `DBI::dbCreateTable`. To use a customized schema, create the schema before calling the function and use `overwrite=FALSE`.

If x is a sf object, the geometry column is converted to binary and written to the database before conversion to geometry/geometry data type. The EPSG code is automatically read from the sf object and used as the SRID.

To override the default path to the bcp command line utility, set the `bcputility.bcp.path` option.

Value

Output from `system2`. See ... to redirect output.

Examples

```
## Not run:
set.seed(33)
x <- data.frame(
  w = rpois(10, 10),
  x = rnorm(10),
  y = sample(LETTERS, 10),
  z = Sys.time()
)
bcpImport(x,
          server = server,
          database = database,
          table = 'mytable')

## End(Not run)
```

Index

bcpExport, 2

bcpImport, 3

system2, 2, 4