Package 'RESS'

February 26, 2015

Type Package

Title Integrates R and Essentia

Version 1.0

Date 2015-02-12

Author Ben Waxer

Maintainer Who to complain to <bwaxer@auriq.com>

Description Query the Essentia Database and access the results in R

License LGPL-3

R topics documented:

`RESS-package` *Integrates R and Essentia.*

Description

essQuery takes a single Essentia command and captures the output in R, where you can save the output to a dataframe or stream it directly into additional analysis.

read.udb takes a file containing any number of Essentia commands and captures the output of the specified statements into R dataframes.

Details

Package:	RESS
Type:	Package
Version:	1.0
Date:	2015-02-24
License:	LGPL-3

Author(s)

Ben Waxer, Data Scientist with Auriq Systems.

Maintainer: Who to complain to <bwaxer@auriq.com>

`essQuery` *essQuery*

Description

Query the UDB database and return the results to R.

Usage

```
essQuery(essentia, aq="", flags="")
```

Arguments

`essentia`	The essentia command to run. The options are "ess task stream category startdate enddate" and "ess task exec". Each stream command can be used to stream any number of files directly into your R analysis. Alternatively, each stream command can save multiple files into separate R dataframes, one file per dataframe. The default value for the essentia argument is "ess task exec".
`aq`	This can be any combination of the aq_tools and standard UNIX commands. However, the output MUST be in a csv format if you want R to capture the output. If you only want to run the command without R capturing the output, add "#Rignore" to the flags argument.
`flags`	Any of the essentia flags can be used here in addition to any of these RESS-specific flags: `#Rignore` : Ignore an 'ess task exec statement. Do not capture the output of the statement into R. `#Rinclude` : Include an 'ess task stream' statement. Capture the output of the statement into R. `#-notitle` : Tell R not to use the first line of the output as the header. `#Rseparate` : Can be used when saving multiple files into an R dataframe using an 'ess task stream' command. Saves each file into a different R dataframe, entitled command1 to commandN, where N is the number of files. `#filelist` : Causes an extra dataframe to be stored in R that saves the list of files streamed into R when streaming multiple files. `#R#name#R#` : Allows any automatically saved dataframe to be renamed to whatever is entered in place of 'name'. This only applies in essQuery when streaming multiple files.

Details

essQuery is used to directly query the database using a single statement. You can call essQuery multiple times to run different statements.

However, you can also use read.udb to read all of the statements in a file instead. Thus if you plan to run multiple statements that may be somewhat related to each other, it is recommended that you use read.udb.

Value

The value returned is the output from querying the database. This can be saved into an R dataframe or directly analyzed in R.

If you use essQuery to save multiple files into a separate R dataframes using a single stream command, the files are stored automatically in R dataframes called command1 to commandN (where N is the number of files) and no value is returned. To change the name of the stored dataframes, use the #R#any_name#R# flag. The dataframes will then be stored as any_name1 to any_nameN.

With #filelist, the extra dataframe is saved as commandN+1 by default, or any_nameN+1 if #R#any_name#R# is also used.

Author(s)

Ben Waxer, Data Scientist with Auriq Systems.

References

See our website at www.auriq.net or our documentation at www.auriq.net/documentation

---

`read.udb` *read.udb*

Description

Read the essentia commands in the stated file, query the UDB database, and save the results into R as dataframes.

Usage

```
read.udb(file, linenumber = "all")
```

Arguments

`file`	The file containing the essentia commands you want to run. 'ess task stream' statements will have their output ignored by R unless there is a '#Rinclude' flag somehwere in the statement line. 'ess task exec' statements will have their output included unless there is a '#Rignore' flag somewhere in the statement line. The file should contain primarily the query commands but can contain the entire essentia script including loading the UDB database if desired. Any command that you want to capture the output from must have its output in csv format.
`linenumber`	This is set to default to all line numbers so that every command in the file is executed. You can specify the line number of the command you wish to run if you dont want to run the entire set of commands in the file.

Details

read.udb reads all of the statements in a file (unless linenumber is specified, see above) and captures the output of the specified commands into R dataframes. By default only the output of 'ess task exec' statements is captured and it's stored in R dataframes command1 to commandN, where N is the number of captured statements.

You can include 'ess task stream' statements by adding a '#Rinclude' flag. This method can be used to stream multiple files into R for data exploration or analysis. Thus if you plan to run multiple statements that may be somewhat related to each other, it is recommended that you use read.udb.

Value

There is no value returned. This command creates a set of R dataframes containing the output from the specified essentia commands in file.

Note

The flags added to the essentia commands in file can include:

`#Rignore` : Ignore an 'ess task exec statement. Do not capture the output of the statement into R.

`#Rinclude` : Include an 'ess task stream' statement. Capture the output of the statement into R.

`#-notitle` : Tell R not to use the first line of the output as the header.

`#Rseparate` : Can be used when saving multiple files into an R dataframe using an 'ess task stream' command. Saves each file into a different R dataframe.

`#filelist` : Causes an extra dataframe to be stored in R that saves the list of files streamed into R when streaming multiple files.

`#R#name#R#` : Allows any automatically saved dataframe to be renamed to whatever is entered in place of 'name'. When used with #Rseparate, saves the files as name1 to nameN, where N is the number of files. Since this still counts as a statement, the next default dataframe saved will be stored as command followed by the number of previous statements run plus one.

Author(s)

Ben Waxer, Data Scientist with Auriq Systems.

References

See our website at www.auriq.net or our documentation at www.auriq.net/documentation.

Index