Package 'RSAP'

January 27, 2015

Version 0.9

Date 2013-02-10

Author Piers Harding <piers@ompka.net>

Maintainer Piers Harding <piers@ompka.net>

Title SAP Netweaver RFC connector for R

Description The SAP Netweaver RFC connector for R

SystemRequirements NW RFC SDK downloaded from http://service.sap.com

Depends R (>= 2.12.0), utils, yaml, reshape

LazyLoad yes

License GPL-3

URL http://github.com/piersharding/RSAP

BugReports http://github.com/piersharding/RSAP/issues

BuildVignettes no

Repository CRAN

Date/Publication 2013-02-21 17:41:36

NeedsCompilation yes

R topics documented:

RSAP-package *SAP RFC Connector for R*

Description

Package **RSAP** implements SAP RFC connectivity for R using the NW RFC SDK

See the package manual for details of installation and use.

The project is hosted at https://github.com/piersharding/RSAP

Details

Enable the use of SAP RFC connectivity to access SAP data, similar to connecting to a database.

con <- RSAPConnect(ashost="nplhost", sysnr="42", client="001", user="developer", passwd="developer", lang="EN", trace="1", lcheck="1")

info = RSAPGetInfo(con) print(info)

parms <- list('BYPASS_BUFFER' = 'X', 'MAX_ENTRIES' = 50, 'TABLE_NAME' = 'T005')

res <- RSAPInvoke(con, "RFC_GET_TABLE_ENTRIES", parms) print(res$ENTRIES) RSAPClose(con)

All RFC table results are returned as data.frame.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPReadTable, RSAPReadCube, RSAPClose

RSAPClose *SAP RFC Close Connections*

Description

Close an open connection to an SAP System

Usage

```
RSAPClose(con)
```

Arguments

con	an Open SAP RFC Conneciton handle

Details

RSAPClose closes an RFC connection to a specified SAP system

Value

Returns true or false

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke

Examples

```
## Not run:
# Close the connection
RSAPClose(con)

## End(Not run)
```

RSAPConnect *SAP RFC Open Connections*

Description

Open connections to an SAP System for RFC calls

Usage

```
RSAPConnect(...)
```

Arguments

...	all SAP connection parameters for the NW RFC SDK

Details

RSAPConnect establishes an RFC connection to the specified SAP system. There are two styles of passing the connection parameters: - RSAPConnect('sap.yml') where the name of A YAML encoded file containing NW RFC SDK connection parameters is passed in - RSAPConnect(ashost = "sap.host.name", user The individual connection parameters are passed as per the requirements of the NW RFC SDK. These parameters are typically: * ashost - the host name of SAP or a SAP Router string * sysnr - The SAP system number - relates to the port or service number * user - SAP login user that is RFC enabled * passwd - user password * lang - login language * lcheck - login check on connection - don't wait until the first call * trace - activate the NW RFC SDK tracing facility - will produce log files

Value

Returns an object that contains the RFC connection object that you can then use to pass to RSAPClose, RSAPInvoke, and RSAPGetInfo.

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPClose, RSAPGetInfo, RSAPInvoke

Examples

```
## Not run:
# full parameter specification
con <- RSAPConnect(ashost="nplhost", sysnr="42", client="001", user="developer", passwd="developer", lang="EN",

# Use a YAML encoded parameter file
con <- RSAPConnect("sap.yml")

## End(Not run)
```

RSAPExecInfoQuery *SAP RFC function calls*

Description

Execute a call to an Info Query, and return a data.frame of the results

Usage

```
RSAPExecInfoQuery(con, infoprovider, infoquery)
```

Arguments

con	an Open SAP RFC Conneciton handle
infoprovider	The technical name of an infoprovider to read
infoquery	The technical name of an infoquery to read

Details

```
con <- RSAPConnect(ashost="nplhost", sysnr="42",
                   client="001", user="developer",
                   passwd="developer", lang="EN",
                   trace="1", lcheck="1")

res <- RSAPExecInfoQuery(con, "0D_NW_M01", "0D_FC_NW_M01_Q0002")

print(res)

RSAPClose(con)
```

Value

Returns a data.frame of the info query results

Note

Use transaction RSRT in SAP to find info providers, and queries.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPReadTable, RSAPClose

Examples

```
## Not run:
# read the NW demo data info cube
  res <- RSAPExecInfoQuery(con, "0D_NW_T01", "20120716",chars=list("0D_NW_SORG", "0D_NW_PROD"), kfigures=list("0

## End(Not run)
```

RSAPGetCube *SAP RFC function calls*

Description

Open connections to an SAP System for RFC calls

Usage

```
RSAPGetCube(con, cube)
```

Arguments

con	an Open SAP RFC Conneciton handle
cube	The technical name of an infocube to read

Details

```
con <- RSAPConnect(ashost="nplhost", sysnr="42",
                   client="001", user="developer",
                   passwd="developer", lang="EN",
                   trace="1", lcheck="1")

res <- RSAPGetCube(con, "0D_NW_T01")

print(res)

RSAPClose(con)
```

Value

Returns a data.frame of the info cube structure information

Note

You can run the RSAPListCubes(con) to get a list of cubes to query.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPReadTable, RSAPClose

Examples

```
## Not run:
# read the NW demo data info cube
  res <- RSAPGetCube(con, "0D_NW_T01")

## End(Not run)
```

RSAPGetInfo *SAP RFC get connection details regarding partner SAP system*

Description

get connection details regarding partner SAP system

Usage

```
RSAPGetInfo(con)
```

Arguments

con	an Open SAP RFC Conneciton handle

Details

RSAPGetInfo get connection details regarding partner SAP system

Value

Returns a named list of details about the connection

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPClose, RSAPInvoke

Examples

```
## Not run:
# Close the connection
info <- RSAPGetInfo(con)
print(info)

## End(Not run)
```

RSAPInvoke *SAP RFC function calls*

Description

Open connections to an SAP System for RFC calls

Usage

```
RSAPInvoke(con, func, parms)
```

Arguments

con	an Open SAP RFC Conneciton handle
func	The name of the SAP RFC function to call
parms	a named list of parameters to pass into the function call

Details

con <- RSAPConnect(ashost="nplhost", sysnr="42", client="001", user="developer", passwd="developer", lang="EN", trace="1", lcheck="1")

info = RSAPGetInfo(con) print(info)

parms <- list('BYPASS_BUFFER' = 'X', 'MAX_ENTRIES' = 50, 'TABLE_NAME' = 'T005')

res <- RSAPInvoke(con, "RFC_GET_TABLE_ENTRIES", parms) print(res$ENTRIES) RSAP-Close(con)

Value

Returns true or false

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPClose

Examples

```
## Not run:
# Close the connection
RSAPInvoke(con, "RFC_FUNCTION_NAME", parms)

## End(Not run)
```

RSAPListCubes *SAP RFC function calls*

Description

List the available BI Cubes from a connected SAP system

Usage

```
RSAPListCubes(con)
```

Arguments

con	an Open SAP RFC Connection handle

Details

```
con <- RSAPConnect(ashost="nplhost", sysnr="42",
                   client="001", user="developer",
                   passwd="developer", lang="EN",
                   trace="1", lcheck="1")

res <- RSAPListCubes(con)

RSAPClose(con)
```

Value

Returns a data.frame of the cube list query

Note

For each cube you can then run RSAPGetCube(con, '<cube name>') to get the details of the cube layout.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPReadTable, RSAPClose

Examples

```
## Not run:
# read the NW demo data info cube
  res <- RSAPListCubes(con)

## End(Not run)
```

RSAPReadCube *SAP RFC function calls*

Description

Open connections to an SAP System for RFC calls

Usage

```
RSAPReadCube(con, cube, ref_date, chars=list(), kfigures=list(), options=list())
```

Arguments

con	an Open SAP RFC Conneciton handle
cube	The technical name of an infocube to read
ref_date	The reference date for data selction from the infocube
chars	list of characteristic technical names that you want in the result set
kfigures	A list of key figure technical names that you want in the result set
options	A list of options and their selection criteria based on the technical names of attributes with a syntax like ABAP SELECT-OPTIONS

Details

```
con <- RSAPConnect(ashost="nplhost", sysnr="42",
                   client="001", user="developer",
                   passwd="developer", lang="EN",
                   trace="1", lcheck="1")

res <- RSAPReadCube(con, "0D_NW_T01", "20120716",
                         chars=list("0D_NW_SORG", "0D_NW_PROD"),
                         kfigures=list("0D_NW_NETV"),
                   options=list(CHANM=list('0D_NW_SORG'),SIGN=list('I'), COMPOP=list('EQ'), LOW=
# or alias
# res <- readCube(con, ...
```

```
print(res)

RSAPClose(con)
```

Value

Returns a data.frame of the info cube query

Note

You can run the RSAPListCubes(con) to get a list of cubes to query. For each cube you can then run RSAPGetCube(con, '<cube name>') to get the details of the cube layout.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPReadTable, RSAPClose

Examples

```
## Not run:
# read the NW demo data info cube
  res <- RSAPReadCube(con, "0D_NW_T01", "20120716",chars=list("0D_NW_SORG", "0D_NW_PROD"), kfigures=list("0D_NW_

## End(Not run)
```

RSAPReadTable *SAP RFC function calls*

Description

Open connections to an SAP System for RFC calls

Usage

```
RSAPReadTable(con, saptable, options=list(), fields=list())
```

Arguments

con	an Open SAP RFC Conneciton handle
saptable	The Data Dictionary name of a table to read
options	list of string values of SQL WHERE clause statements to apply to the table select
fields	A list of column names that you want returned from the table

Details

```
con <- RSAPConnect(ashost="nplhost", sysnr="42",
                   client="001", user="developer",
                   passwd="developer", lang="EN",
                   trace="1", lcheck="1")

res <- RSAPReadTable(con, "SFLIGHT2")
# or use alias
# res <- readTable(con, "SFLIGHT2")

print(res)

RSAPClose(con)
```

Value

Returns a data.frame of the table contents selected

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke, RSAPClose

Examples

```
## Not run:
# read the flight data demo table
res <- RSAPReadTable(con, "SFLIGHTS2", options=list("CARRID = 'AA'"), fields=list('CARRID', 'CONNID', 'FLDATE',

## End(Not run)
```

RSAPValidHandle *SAP RFC check valid connection*

Description

Check a connection handle is valide

Usage

```
RSAPValidHandle(con)
```

Arguments

con	an SAP RFC Connection handle

Details

RSAPValidHandle check a connection handle to SAP is valid

Value

Returns true or false

Note

Not much to note here.

Author(s)

Piers Harding

See Also

RSAPConnect, RSAPGetInfo, RSAPInvoke

Examples

```
## Not run:
# Close the connection
RSAPValidHandle(con)

## End(Not run)
```

Index