

# Package ‘bayeos’

July 22, 2013

**Type** Package

**Title** BayEOS Server Access

**Version** 1.4.1

**Date** 2013-07-22

**Author** Stefan Holzheu, Oliver Archner

**Maintainer** Stefan Holzheu <stefan.holzheu@bayceer.uni-bayreuth.de>

**Description** This package provides functions to access BayEOS databases

**Imports** RCurl, XML, zoo, methods

**License** LGPL-3

**LazyLoad** yes

## R topics documented:

bayeos-package . . . . .	2
bayeos.acl . . . . .	2
bayeos.connect . . . . .	4
bayeos.data.frame . . . . .	5
bayeos.references . . . . .	7
bayeos.timeseries . . . . .	8
bayeos.tree . . . . .	10
<b>Index</b>	<b>14</b>

---

bayeos-package                    *Package to access BayEOS data from R via XMLRPC*

---

### Description

The packages provides the user with an easy to use frontend to access a BayEOS database from R. The package is based on a modified XMLRPC package of the omega project for statistical computing ([www.omegahat.org](http://www.omegahat.org)).

### Author(s)

Stefan Holzheu <[stefan.holzheu@uni-bayreuth.de](mailto:stefan.holzheu@uni-bayreuth.de)> Oliver Archner <[oliver.archner@uni-bayreuth.de](mailto:oliver.archner@uni-bayreuth.de)>

### See Also

[bayeos.connect](#) [bayeos.getSeries](#) [bayeos.cd](#) [bayeos.getACL](#) [bayeos.getDF](#)

### Examples

```
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet', 'gast', 'gast')
data = bayeos.getSeries(c(2717,2718))
plot(data)
bayeos.close()
```

---

bayeos.acl                    *Manage ACL-Lists of BayEOS Tree*

---

### Description

Functions to manage Access Control Lists of BayEOS-Nodes.

### Usage

```
bayeos.getACL(id=NA,ls=FALSE,con=1)
bayeos.createACL(rid,id=NA,read=TRUE,write=FALSE,exec=FALSE,inherit=FALSE,ls=FALSE,con=1)
bayeos.deleteACL(rid,id=NA,ls=FALSE,con=1)
bayeos.updateACL(right,value,rid,id=NA,ls=FALSE,con=1)
bayeos.getRoles(con=1)
bayeos.getRid(name,con=1)
```

**Arguments**

con	connection identifier returned by bayeos.connect
id	numeric node id or NA for current folder
rid	numeric role id or character role name
ls	Print out ACL
right	character one of read,write,exec,inherit
value	boolean value of specified right
read,write,exec,inherit	boolean value of specified right
name	User or group name to search for

**Details**

BayEOS offers a flexible access control list.

bayeos.getRid will search the bayeos.getRoles data frame for the given name. It will first search the unname column. When there is no result it will continue with the fullname column.

**Value**

bayeos.getACL,bayeos.getRoles returns a data frame. bayeos.createACL, bayeos.deleteACL, bayeos.updateACL return success as boolean. bayeos.getRid return the numeric role id.

**Author(s)**

S. Holzheu, O. Archner

**Examples**

```
# open connection
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast')

# show the current folder
dir=bayeos.ls()

bayeos.getACL()
bayeos.getACL(dir$id[1])

bayeos.close()
```

---

bayeos.connect	<i>Create and close a BayEOS connection from R</i>
----------------	--

---

### Description

This function creates an BayEOS connection via XMLRPC. It returns an integer of the connection index. Normally all operations run per default on the first connection. It is possible to handle multiple BayEOS connections. Use different connection numbers in this case.

### Usage

```
bayeos.connect(url=NULL, user=NULL, password=NULL, save_as=NULL,
force_interactive=FALSE, con=1, .debugRPC=FALSE)
bayeos.getConnection(con=1)
bayeos.close(con=1)
bayeos.call(con, method, . . . , .convert=TRUE)
```

### Arguments

url	the URL to the BayEOS servlet or alias of the .bayeos.pwd-file
user	BayEOS login
password	BayEOS password
force_interactive	Setting TRUE will prompt for login and password
save_as	If set, will save the connection data in the .bayeos.pwd-file
con	connection number
method	XMLRPC method
. . .	XMLRPC arguments
.debugRPC	flag if TRUE each call shows the XMLRPC Strings
.convert	flag if TRUE XMLRPC response is converted to R objects

### Details

Normally one does not want to handle clear text passwords in scripts. To avoid this, it is recommended to use the save\_as flag to create a simple text-file ".bayeos.pwd" in the home drive of the user. It will contain one line per connection in the format "alias url user encrypted\_password". Together with the automatically created ".bayeos.pwd\_key" it is possible to decrypt the passwords. So please make sure, that these two files are not readable to anyone else!

### Value

If login is successful it returns an integer of the connection index. Otherwise it throws a XML-RPC exception. bayeos.getConnection returns the connection list. bayeos.call runs a XMLRPC request on a connection and returns the result (developer and debugging function)

**Author(s)**

S. Holzheu, O. Archner

**See Also**[bayeos.getSeries](#)  
[bayeos.cd](#)**Examples**

```
## Not run:
# Open a connection to the given url with login and password 'gast'
con=bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast')
# Open a connection to the given url with login and password 'gast'
# and stores the connection data in the .bayeos.pwd-file in the
# HOME-directory of the user
con=bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast',
save_as='bayceer_gast')
# Open a connection using the .bayeos.pwd file
# will look for a configuration line 'bayceer_gast'
con=bayeos.connect('bayceer_gast')
# will take the first line of the .bayeos.pwd-file
con=bayeos.connect()
# will take the first line of the .bayeos.pwd-file and store the connection
# informationen as a second entry in the .bayeoscon list variable
con=bayeos.connect(con=2)
# close connection
bayeos.close()
# close connection number 2
bayeos.close(2)

## End(Not run)
```

---

`bayeos.data.frame`*Read and write data.frames using a bayeos server connection*

---

**Description**

Functions to handle data.frames with a bayeos server. (BayEOS-Server Version &gt;= 1.9)

**Usage**

```
bayeos.getDF(ids,rowNr=NULL,con=1)
bayeos.getDFZoo(ids,zooIndexColumn=1,aggregateFun=NULL,con=1)
bayeos.writeDF(ids,data,rowNr=NULL,con=1)
bayeos.importDF(id,data,columnSelection=NULL,columnDestination=NULL,append=FALSE,con=1)
```

**Arguments**

con	connection identifier returned by bayeos.connect
ids	data.frame or data.frame.column ids
id	data.frame id or name
data	a data.frame to be imported
rowNr	a integer vector of row numbers
columnDestination	character vector of destination data.frame columns. Will overwrite column names from data
columnSelection	column numbers starting with 1 or character vector of column names to be imported
append	logical, with append=FALSE allways whole column is written
zooIndexColumn	Column number of data.frames to take as zoo index column.
aggregateFun	aggregateion function name e.g. mean

**Value**

bayeos.getDF returns a data.frame. bayeos.getDFZoo returns a zoo object bayeos.writeDF and bayeos.importDF return a logical vector for every column with the write success

**Author(s)**

S. Holzheu, O. Archner

**Examples**

```
## Not run:
# open connection
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast')

# will not work with guest connection
id=bayeos.createDF('testDF')
bayeos.cd(id)
data=data.frame(sp1=logical(10),
                sp2=1:10,
                sp3=rnorm(10),
                sp4=as.POSIXct(1:10,origin='2013-01-01'),
                sp5=as.character(101:110)
                )
bayeos.importDF(data)
bayeos.getDF(id)
dir=bayeos.ls()
bayeos.writeDF(dir$id[2],data.frame(1:20))
bayeos.writeDF(dir$id[5],data.frame(as.character(1:5)))
bayeos.getDF(id)
bayeos.cd()
```

```

bayeos.deleteNode(id,confirm=FALSE)

# close connection
bayeos.close()

## End(Not run)

```

---

bayeos.references      *Manage References in BayEOS Tree*

---

## Description

Functions to get, create or delete references.

## Usage

```

bayeos.getReferences(id,con=1)
bayeos.createReferences(id,to,tree,create=FALSE,id_parent=NULL,con=1)
bayeos.deleteReferences(id,type,con=1)
bayeos.createReferenceNode(name,tree,id_parent=NULL,con=1)
bayeos.deleteReferenceNode(id,con=1)

```

## Arguments

con	connection identifier returned by bayeos.connect
id	numeric object id
tree	Tree to search in. One of 'device','compartment','location','target','unit','web'
to	Object name or id to create a reference
create	When true, createReferences will call createReferenceNode when to is not found
id_parent	Parent Node. When Null the root node is taken
name	Name of the references Node to create.
type	Type of the referenced object; use 'all' to delete all direct references.

## Value

returns a data frame with the columns name,type,direct

## Author(s)

S. Holzheu, O. Archner

## Examples

```
## Not run:
# open connection
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet', 'gast', 'gast')

# create folder and series (only works, when rights are sufficient)
newFolder=bayeos.createFolder('test folder')
bayeos.createReferences(newFolder, 'Somewhere', 'Location')
newNode=bayeos.createSeries('test series')
bayeos.createReferences(newNode, 'Celsius', 'Unit')
bayeos.deleteNode(newNode)
# close connection
bayeos.close()

## End(Not run)
```

---

bayeos.timeseries      *Read and write time series data using a bayeos server connection*

---

## Description

bayeos.getSeries creates a zoo object from a BayEOS connection. bayeos.readCSV reads data from csv file into a zoo object. bayeos.import and bayeos.writeSeries write zoo objects or data frames to BayEOS database.

## Usage

```
bayeos.getSeries(ids, con=1, from='yesterday', until='today', interval=NA,
aggfunc=NA, aggint=NA, maxrows=10000,
csFlag=FALSE, statusFilter=c(0,1,2,3,4,8,9))
bayeos.readCSV(file, tz='Etc/GMT-1', dateformat=c('%d.%m.%Y %H:%M'), skip=0, sep=';', dec='.',
na.strings='NA', header=TRUE, comment.char="#", skipCols=0, ...)
bayeos.import(data, folder='./', columnSelection=NULL, columnDestination=NULL, con=1,
update=FALSE, status=NA)
bayeos.writeSeries(ids, data, con=1, update=FALSE, status=NA)
bayeos.setStatus(ids, datetimes, status, con=1)
bayeos.zoo2df(data)
```

## Arguments

ids	time series ids
con	connection index returned by bayeos.connect
data	time series data as zoo object



from	datetime string or keyword 'yesterday'
until	datetime string or keyword 'today'
interval	string of the following keywords: today, this week, this month, this year, yesterday, last week, last month, last year. Overwrites from and until settings.
aggfunc	aggregation function; valid values see bayeos.getConnection(con)\$aggfunc
aggint	aggregation interval; valid values see bayeos.getConnection(con)\$aggint
maxrows	maximum number of rows.
csFlag	dataframe will contain additional columns with status or counts
statusFilter	Only query values with status IDs within this vector
update	setting to TRUE will delete existing data from min(data\$datetime) until max(data\$datetime). Otherwise data of existing timestamps is silently ignored. Use with care!
columnDestination	character vector of destination time series. Will overwrite column names from data
columnSelection	column numbers starting with 1 or character vector of column names to be imported
datetimes	Vector of POSIXct datetimes
status	Integer or character; valid values see bayeos.getConnection(con)\$status
...	Further arguments to be passed to 'read.table'.

## Details

BayEOS-R package handles time series data as zoo data type. See package [zoo](#) for more information.

bayeos.getSeries will return a zoo object containing one or more time series. Search parameters from and until should be given as character strings. For POSIXct arguments as.character() will be called. 'from' and 'until' are interpreted in the local time zone of the series. The POSIXct time zone of the zoo object is set to the time zone of the the series. When there are time series with different time zones in the zoo object the time zone of the first series is relevant.

bayeos.readCSV is a wrapper for the read.table function of R. It returns a zoo object. It support separated date and time columns. For dateformat description see [strftime](#). Further arguments conform to [read.table](#).

bayeos.import and bayeos.writeSeries write data to the database. bayeos.import will operate in a folder creating the time series objects if not existing. bayeos.writeSeries writes data to a existing time series.

## Value

bayeos.getSeries, bayeos.readCSV return a time series data as zoo object. bayeos.import, bayeos.writeSeries and bayeos.setStatus return TRUE on success. bayeos.zoo2df transforms the zoo data object to data frame type used in the first releases of the package.

**Author(s)**

S. Holzheu, O. Archner

**See Also**

[bayeos.cd](#)  
[bayeos.connect](#)

**Examples**

```
## Not run:
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast')
bayeos.getSeries(2718)
bayeos.getSeries(c(2718,2717))
bayeos.getSeries(2718,from='2006-07-01 00:00:00',until='2006-07-01 01:00:00')
bayeos.getSeries(2718,from='2006-01-01 00:00:00',until='2006-02-01 01:00:00',aggfunc='Avg',aggint='day')
bayeos.getSeries(2718,interval='Last Year',aggfunc='Avg',aggint='day')

# Read data from csv
f = system.file("sampleData", "datetime.csv", package = "bayeos")
bayeos.readCSV(f,dec=',',sep=';',na.string='<NA')
f = system.file("sampleData", "date_time.csv", package = "bayeos")
bayeos.readCSV(f,dec=',',sep=';',dateformat=c('%d.%m.%Y','%H:%M'),na.string='<NA',tz='Etc/GMT-8')

# Write the data in timeseries of the current folder
# Series names are taken from data (this will not run with the guest connection!)
f = system.file("sampleData", "datetime.csv", package = "bayeos")
data=bayeos.readCSV(f,dec=',',sep=';',na.string='<NA')
bayeos.import(data,status='valid')

bayeos.close()

## End(Not run)
```

---

 bayeos.tree

---

*Manage BayEOS Timeseries Tree*


---

**Description**

Functions to list current folder, change folder and to create new folder or series

**Usage**

```
bayeos.ls(con=1,full=FALSE)
bayeos.cd(id=NULL,con=1,ls=TRUE)
bayeos.createFolder(name,id_parent=NULL,con=1,ls=TRUE)
bayeos.createSeries(name,con=1,ls=TRUE,warningLevel=1,tz='Etc/GMT-1',description='',
```

```

resolution=600,planstart=NULL,planend=NULL,intervaltype='undefined',id_parent=NULL)
bayeos.createDFColumn(name,colNr,class,description=NULL,con=1,id_parent=NULL,ls=TRUE,
planstart=NULL,planend=NULL,recstart=NULL,recend=NULL)
bayeos.createDF(name,description=NULL,tz=NULL,con=1,id_parent=NULL,ls=TRUE,
planstart=NULL,planend=NULL,recstart=NULL,recend=NULL)
bayeos.updateDF(id,name=NA,description=NA,tz=NA,con=1,id_parent=NA,ls=FALSE,
planstart=NA,planend=NA,recstart=NA,recend=NA)
bayeos.updateDFColumn(id,name=NA,colNr=NA,class=NA,description=NA,con=1,id_parent=NA,ls=FALSE,
planstart=NA,planend=NA,recstart=NA,recend=NA)
bayeos.updateSeries(id,name=NA,tz=NA,description=NA,resolution=NA,
planstart=NA,planend=NA,intervaltype=NA,id_parent=NA,con=1,ls=FALSE)
bayeos.updateNode(id,name=NA,id_parent=NA,ls=TRUE,con=1)
bayeos.deleteNode(id,con=1,confirm=TRUE,ls=TRUE,removeRows=FALSE)
bayeos.getChilds(id,isExtern=FALSE,maxDepth=-1,pathFilter='',classFilter='',con=1)
bayeos.find(search,tree='Folder',con=1)
bayeos.getNode(id,con=1)

```

### Arguments

con	connection identifier returned by bayeos.connect
name	name of the series or folder to create
id	numeric folder id or path character, NULL for root folder
id_parent	numeric folder id or path character
ls	print out modified directory
full	flag, if FALSE ls() returns short attribute columns
maxDepth	interger, max. depth for tree search, -1==unlimited
pathFilter	Filter string for ANT-like tree search. See below for details.
search	String to search nodes. For substring search use '*temp*'
classFilter	Object type filter string
tree	Tree to search in. One of 'folder','device','compartment','location','target','unit','web'
planstart	Planned start of the data. Date time string interpreted in the local time zone of the series.
planend	Planned end of the data. Date time string interpreted in the local time zone of the series.
recstart	Record start of the data. Date time string interpreted in the local time zone of the data.
recend	Record end of the data. Date time string interpreted in the local time zone of the data.
description	Node description
resolution	time series resolution in seconds
tz	time zone string e.g. Etc/GMT-1

intervaltype	one of undefined,start,end
colNr	Data frame column number
class	Data frame column data type (STRING,BOOLEAN,INTEGER,DOUBLE,DATE)
confirm	Needs confirmation before deleting a node
warningLevel	Checks meta data of node. 0=suppress warnings
removeRows	if true, a removeAllRows is called on the node before deleting
isExtern	set to TRUE to search the web object tree

### Value

bayeos.ls prints out the current directory and returns an invisible data.frame with columns id,class,name,rec\_start,rec\_end.  
 bayeos.createXXX return ID of new node. bayeos.updateXXX return true on success bayeos.getChildren and bayeos.find returns a data frame with the columns id,class,name,path

### Author(s)

S. Holzheu, O. Archner

### Examples

```
## Not run:
# open connection
bayeos.connect('http://www.bayceer.uni-bayreuth.de/servlet/XMLServlet','gast','gast')

# show the current folder
bayeos.ls()

# change to a subfolder
bayeos.cd(14333)
# change to a folder
bayeos.cd(36076)
# change to the parent folder
bayeos.cd('..')
# change to root folder
bayeos.cd()
# change to folder via path
bayeos.cd('Micrometeorology Dept/AWS Ecological Botanical Garden')

## Childs
# get all childs of a specified node
bayeos.getChildren(83)
# get all childs of a specified node using filter
# ** is a wildcard for any character in the path including the folder separator
# * is a wildcard for all characters except the folder separator
# To find all Nodes beginning with "temperature" anywhere in the tree
bayeos.getChildren(83,pathFilter='**/temperature*')

## Searching the trees
# Searching all nodes having temperature somewhere in the name
```

```
bayeos.find('*temperatur*')
# Finding Unit 'g/l'
bayeos.find('m/s', tree='Unit')
# getting all units
bayeos.find('*', tree='Unit')

# create folder and series (only works, when rights are sufficient)
newFolder=bayeos.createFolder('test folder')
bayeos.createReferences(newFolder, 'Somewhere', 'Location')
newNode=bayeos.createSeries('test series')
bayeos.createReferences(newNode, 'Celsius', 'Unit')
bayeos.deleteNode(newNode)
# close connection
bayeos.close()

## End(Not run)
```

# Index

- \*Topic **acl**
  - bayeos.acl, 2
- \*Topic **connection**
  - bayeos.connect, 4
- \*Topic **data.frame**
  - bayeos.data.frame, 5
- \*Topic **data**
  - bayeos.timeseries, 8
- \*Topic **meta data**
  - bayeos.references, 7
- \*Topic **package**
  - bayeos-package, 2
- \*Topic **references**
  - bayeos.references, 7
- \*Topic **timeseries**
  - bayeos-package, 2
  - bayeos.timeseries, 8
- \*Topic **tree**
  - bayeos.tree, 10

bayeos (bayeos-package), 2

bayeos-package, 2

bayeos.acl, 2

bayeos.call (bayeos.connect), 4

bayeos.cd, 2, 5, 10

bayeos.cd (bayeos.tree), 10

bayeos.close (bayeos.connect), 4

bayeos.connect, 2, 4, 10

bayeos.createACL (bayeos.acl), 2

bayeos.createDF (bayeos.tree), 10

bayeos.createDFColumn (bayeos.tree), 10

bayeos.createFolder (bayeos.tree), 10

bayeos.createReferenceNode (bayeos.references), 7

bayeos.createReferences (bayeos.references), 7

bayeos.createSeries (bayeos.tree), 10

bayeos.data.frame, 5

bayeos.deleteACL (bayeos.acl), 2

bayeos.deleteNode (bayeos.tree), 10

bayeos.deleteReferenceNode (bayeos.references), 7

bayeos.deleteReferences (bayeos.references), 7

bayeos.find (bayeos.tree), 10

bayeos.getACL, 2

bayeos.getACL (bayeos.acl), 2

bayeos.getChilds (bayeos.tree), 10

bayeos.getConnection (bayeos.connect), 4

bayeos.getDF, 2

bayeos.getDF (bayeos.data.frame), 5

bayeos.getDFZoo (bayeos.data.frame), 5

bayeos.getNode (bayeos.tree), 10

bayeos.getReferences (bayeos.references), 7

bayeos.getRid (bayeos.acl), 2

bayeos.getRoles (bayeos.acl), 2

bayeos.getSeries, 2, 5

bayeos.getSeries (bayeos.timeseries), 8

bayeos.import (bayeos.timeseries), 8

bayeos.importDF (bayeos.data.frame), 5

bayeos.ls (bayeos.tree), 10

bayeos.readCSV (bayeos.timeseries), 8

bayeos.references, 7

bayeos.setStatus (bayeos.timeseries), 8

bayeos.timeseries, 8

bayeos.tree, 10

bayeos.updateACL (bayeos.acl), 2

bayeos.updateDF (bayeos.tree), 10

bayeos.updateDFColumn (bayeos.tree), 10

bayeos.updateNode (bayeos.tree), 10

bayeos.updateSeries (bayeos.tree), 10

bayeos.writeDF (bayeos.data.frame), 5

bayeos.writeSeries (bayeos.timeseries), 8

bayeos.zoo2df (bayeos.timeseries), 8

read.table, 9

strftime, 9

zoo, 9