Package 'largeList'

April 17, 2017

Type Package

Title Serialization Interface for Large List Objects

Version 0.3.1

Date 2017-04-17

Author Yuchun Zhang

Maintainer Yuchun Zhang <yczhangvrc@gmail.com>

Depends R (>= 3.2.0)

Description Functions to write or append a R list to a file, as well as read, remove, modify elements from it without restoring the whole list.

License GPL (>= 2)

RoxygenNote 6.0.1

BugReports https://github.com/Yuchun-Zhang/R_largeList/issues

URL https://github.com/Yuchun-Zhang/R_largeList

VignetteBuilder knitr

Suggests testthat, futile.logger, knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-04-17 21:41:14 UTC

R topics documented:

`getList` *Create a R object and bind with file.*

Description

Create a R object and bind with file.

Usage

```
getList(file, compress = TRUE, verbose = FALSE, truncate = FALSE)
```

Arguments

`file`	Name of file
`compress`	`TRUE/FALSE` Use compression for elements or not.
`verbose`	`TRUE/FALSE` Print extra info
`truncate`	`TRUE/FALSE` Truncate the file or not.

Details

Create a R object of class "largeList" and bind it with a file. If the file exists, the R object will be bound to the file. If the file does not exist or `truncate == TRUE`, an empty list will be written into the given file and then bind it with R object. Later the R object can be used as a normal list and all the manipulation will be done within the file binding to it.

Value

A R object of class "largeList"

See Also

`largeList`

Examples

```
largelist_object <- getList("example.llo", verbose = TRUE, truncate = TRUE)
```

`getListLength` *Get number of elements in a list file.*

Description

Get number of elements in a list file.

Usage

```
getListLength(file)
```

Arguments

`file`	Name of file.

Value

An integer.

See Also

`largeList`

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")
getListLength(file = "example.llo")
```

`getListName` *Get names of elements in a list file.*

Description

Get names of elements in a list file.

Usage

```
getListName(file)
```

Arguments

`file`	Name of file.

Value

A charater vector.

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")
getListName(file = "example.llo")
```

---

isListCompressed *Check if elements are compressed in the file.*

Description

Check if elements are compressed in the file.

Usage

```
isListCompressed(file)
```

Arguments

file	Name of file.

Value

TRUE/FALSE

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo", compress = FALSE)
isListCompressed(file = "example.llo") # get FALSE
```

largeList *largeList: Serialization Interface for Large List Objects*

Description

Functions to write or append a R list to a file, read, modify or remove elements from it without restoring the whole list.

Details

R objects will be serialized with an uncompressed/ compressed (zlib, default level) non-ascii little-endian format, which is similar to `saveRDS`. Two ordered tables are created at the end of data for quick lookups, one for indices and one for element names. Notice that, all the names will be truncated to 16 characters.

Given indices or names of elements, positions will be directly extracted or extracted via binary search within the name-position table. Then required elements are located and unserialized. Therefore it will not restore the whole list into memory.

With overloads of operators, list objects stored in files can be manipulated as simply as the normal list objects in R.

In the current version, only basic data types are supported, including NULL, integer, numeric, character, complex, raw, logic, factor, list, matrix, array and data.frame. Types like function, data.table are not supported.

Supported maximum size of R objects is `2^31 -1`, supported maximum file size is `2^63 -1` bytes.

Following functions are provided:

- `saveList` Save or append elements to a list file.
- `readList` Get elements from a list file.
- `removeFromList` Remove elements from a list file.
- `modifyInList` Modify elements in a list file.
- `modifyNameInList` Modify names of elements in a list file.
- `getListName` Get number of elements in a list file.
- `getListLength` Get names of elements in a list file.

Some operators / functions are overloaded.

- `getList` Bind a R object with a list file.
- `[.largeList` Get elements.
- `[[.largeList` Get element.
- `$.largeList` Get element, same as `[[.largeList`, no partial matching.

- `[<-.largeList` If index provided, it modifies, appends or removes the elements with given indices, otherwise it appends value to list.
- `[[<-.largeList` If index provided, it modifies, appends or removes the element with given index, otherwise it saves value to list.
- `$<-.largeList` Same as `[[<-.largeList`, no partial matching.
- `length.largeList` Get length of list stored in file.
- `length<-.largeList` Set length of list stored in file.
- `names.largeList` Get names of elements stored in file.
- `names<-.largeList` Set names of elements stored in file.

`length.largeList` *Overload of function length.*

Description

Get the length of list stored in file.

Usage

```
## S3 method for class 'largeList'
length(x)
```

Arguments

x	A largeList object created by `getList`.

Value

An integer.

See Also

`largeList`

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
length(largelist_object) ## get 3
```

length<-.largeList *Overload of function length<-.*

Description

Set the length of list stored in file.

Usage

```
## S3 replacement method for class 'largeList'
length(x) <- value
```

Arguments

x	A largeList object created by getList.
value	An integer number

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
length(largelist_object) <- 5 ## get list("A" = 1, "B" = 2, "C" = 3, NULL, NULL)
```

modifyInList *Modify elements in a list file.*

Description

Modify elements in a list file.

Usage

```
modifyInList(file, index, object)
```

Arguments

file	Name of file.
index	A numeric, logical or character vector.
object	A list consisting of replacement values.

Details

It modifies elements with given indices by replacement values provided in parameter object. If length of replacement values is shorter than length of indices, values will be used circularly. This function may relocate all the data in the stored file, thus can be very slow! Please consider to call this function batchwise instead of one by one.
When it takes long time to process, some verbose info will be printed to console, which can be switched off by setting `options(list(largeList.report.progress = FALSE))`.

Value

invisible `TRUE` if no error occurs.

See Also

`largeList`

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")

# by numeric indices
modifyInList(file = "example.llo", index = c(1,2), object = list("AA","BB"))

# by names
modifyInList(file = "example.llo", index = c("AA","BB"), object = list("A","B"))

# by logical indices
modifyInList(file = "example.llo", index = c(TRUE, FALSE, TRUE), object = list("A","B"))
```

`modifyNameInList` *Modify names of elements in a list file.*

Description

Modify names of elements in a list file.

Usage

```
modifyNameInList(file, index, name)
```

Arguments

`file`	Name of file.
`index`	A numeric or logical vector.
`name`	A character vector consisting replacement names.

Details

Modify names of elements with given indices by replacement values provided in parameter name. If the length of replacement values is shorter than the length of indices, values will be used circularly.

Value

invisible TRUE if no error occurs.

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")

# by numeric indices
modifyNameInList(file = "example.llo", index = c(1,2), name = c("AA","BB"))

# by logical indices
modifyNameInList(file = "example.llo", index = c(TRUE, TRUE, FALSE), name = c("AA","BB"))
```

---

names.largeList *Overload of function names.*

Description

Get the names of elements stored in list file.

Usage

```
## S3 method for class 'largeList'
names(x)
```

Arguments

x	A largeList object created by getList.

Value

A character vector.

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
names(largelist_object) ## get c("A", "B", "C")
```

`names<-.largeList` *Overload of operator names<-.*

Description

Overload of operator names<-.

Usage

```
## S3 replacement method for class 'largeList'
names(x) <- value
```

Arguments

`x`	A largeList object created by `getList`.
`value`	A character vector.

Details

It behaviours the same as a normal list object.

See Also

`largeList`

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
names(largelist_object) <- c("AA", "BB", "CC")
```

print.largeList *Overload of function print.*

Description

If length of list dose not exceed getOption("max.print"), all elements will be printed, otherwise, elements beyond getOption("max.print") will be omitted.

Usage

```
## S3 method for class 'largeList'
print(x, ...)
```

Arguments

x	A largeList object created by getList.
...	arguments to be passed.

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
print(largelist_object) ## print to screen
largelist_object ## print to screen
```

readList *Get elements from a list file.*

Description

Get elements from a list file.

Usage

```
readList(file, index = NULL)
```

Arguments

file	Name of file.
index	NULL or a numeric, character, logical vector.

Details

If no indices provided, the whole list will be read. Given index could be a numeric (integer) vector, a logical vector or a character vector representing the names. If there exist more then one elements corresponding to a given name, the first matched will be returned (not necessary to be the first one in index order). If there are no elements with given name, NULL will be returned.
Files created by saveRDS can't be read.
When it takes long time to process, some verbose info will be printed to console, which can be switched off by setting `options(list(largeList.report.progress = FALSE))`.

Value

A list object.

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")

# read the whole list
readList(file = "example.llo")

# by numeric indices
readList(file = "example.llo", index = c(1, 3))

# by names
readList(file = "example.llo", index = c("A", "B"))

# by logical indices
readList(file = "example.llo", index = c(TRUE, FALSE, TRUE))
```

---

removeFromList *Remove elements from a list file.*

Description

Remove elements from a list file.

Usage

```
removeFromList(file, index)
```

Arguments

file	Name of file.
index	A numeric, logical or character vector.

Details

It removes elements with given indices or names. This function may relocate all the data in the stored file, thus can be very slow! Please consider to call this function batchwise instead of one index by one index.
When it takes long time to process, some verbose info will be printed to console, which can be switched off by setting `options(list(largeList.report.progress = FALSE))`.

Value

invisible `TRUE` if no error occurs.

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))
saveList(object = list_1, file = "example.llo")

# by numeric indices
removeFromList(file = "example.llo", index = c(2))

# by name
removeFromList(file = "example.llo", index = c("A"))

# by logical indices
removeFromList(file = "example.llo", index = c(TRUE))
```

---

saveList *Save or append elements to a list file.*

Description

Save or append elements to a list file.

Usage

```
saveList(object, file, append = FALSE, compress = TRUE)
```

Arguments

object	A list object to save or append.
file	Name of file.
append	TRUE/FALSE, TRUE refers to truncating and saving. FALSE refers to appending.
compress	TRUE/FALSE, using compression or not.

Details

Save or append a list (with or without names) to a file. Notice that, all names will be truncated to 16 characters. Rest attributes of lists will be discarded.
Files generated by this function are not readable by readRDS.
When it takes long time to process, some verbose info will be printed to console, which can be switched off by setting options(list(largeList.report.progress = FALSE)).

Value

invisible TRUE if no error occurs.

See Also

largeList

Examples

```
list_1 <- list("A" = c(1,2), "B" = "abc", list(1, 2, 3))

# save list_1 to file using compression.
saveList(object = list_1, file = "example.llo", append = FALSE, compress = TRUE)

# append list_1 to file, compress option will be extracted from the file.
saveList(object = list_1, file = "example.llo", append = TRUE)
```

[.largeList *Overload of operator [].*

Description

Overload of operator [].

Usage

```
## S3 method for class 'largeList'
x[index = NULL]
```

Arguments

x	A largeList object created by getList.
index	A numeric, logical or character vector.

Details

It behaviours the same as a normal list object.

Value

A list

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
largelist_object[c(1, 2)] ## get list("A" = 1, "B" = 2)
largelist_object[c(TRUE, FALSE, TRUE)] ## get list("A" = 1, "C" = 3)
largelist_object[c("A", "C")] ## get list("A" = 1, "C" = 3)
```

[<-.largeList *Overload of operator []<-.*

Description

Overload of operator []<-.

Usage

```
## S3 replacement method for class 'largeList'
x[index = NULL] <- value
```

Arguments

x	A largeList object created by getList.
index	NULL or a numeric, logical, character vector.
value	NULL, a vector or a list.

Details

It behaviours the same as a normal list object. If index is not provided, elements in value will be appended to the list file. If value is NULL, elements with given indices will be removed.

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[] <- list("A" = 1, "B" = 2)  ## append list to the list file
largelist_object[] <- list("C" = 3, "D" = 4)  ## append list to the list file
largelist_object[1] <- NULL ## remove first element
largelist_object["B"] <- NULL ## remove element with name "B"
largelist_object[c("C","D")] <- c(5, 6) ## change value
largelist_object[2] <- 5 ## change value
largelist_object[c(4, 5)] <- list(6, 7) ## append 6, 7 to 4th, 5th position and NULL to 3rd position
```

[[.largeList *Overload of operator [[]].*

Description

Overload of operator [[]].

Usage

```
## S3 method for class 'largeList'
x[[index = NULL]]
```

Arguments

x	A largeList object created by getList.
index	A numeric or character vector of length 1.

Details

It behaviours the same as a normal list object.

Value

A R object.

See Also

largeList

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
largelist_object[[1]] ## get 1
largelist_object[["B"]] ## get 2
```

`[[<-.largeList` *Overload of operator [[]]<-.*

Description

Overload of operator [[]]<-.

Usage

```
## S3 replacement method for class 'largeList'
x[[index = NULL]] <- value
```

Arguments

`x`	A largeList object created by `getList`.
`index`	`NULL` or a numeric, character vector with length 1.
`value`	`NULL`, a vector or a list.

Details

It behaviours the same as a normal list object. If index is not provided, the list file binding with the largeList object will be truncated and elements in value will be saved to the list file. If value is `NULL`, element with given index will be removed.

See Also

`largeList`

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("A" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
largelist_object[[1]] <- NULL ## remove first element
largelist_object[["B"]] <- NULL ## remove element with name "B"
largelist_object[["C"]] <- 5 ## change value
largelist_object[[2]] <- 5 ## change value
largelist_object[[4]] <- 6 ## append 6 to 4th and NULL to 3rd position
```

`$.largeList` *Overload of operator $.*

Description

Overload of operator $.

Usage

```
## S3 method for class 'largeList'
x$index
```

Arguments

`x`	A largeList object created by `getList`.
`index`	A character vector of length 1.

Details

It behaviours different from the list object in R. Here `x$name` is equivalent to `x[["name"]]`, **no partial matching**.

Value

A R object.

See Also

`[[.largeList` `largeList`

Examples

```
largelist_object <- getList("example.llo", truncate = TRUE)
largelist_object[[]] <- list("AA" = 1, "B" = 2, "C" = 3)  ## assign list to the list file
largelist_object$B ## get 2
largelist_object$A ## get NULL, not 1 from "AA" since no partial matching happens.
```

`$<-.largeList` *Overload of operator $<-.*

Description

Overload of operator $<-.

Usage

```
## S3 replacement method for class 'largeList'
x$index <- value
```

Arguments

`x`	A largeList object created by `getList`.
`index`	`NULL`, a numeric or character vector with lenght 1.
`value`	`NULL`, a vector or a list.

Details

x$A <- 1 is equivalent to x[["A"]] <- 1

See Also

`[[<-.largeList` `largeList`

Index