Package 'osum'

May 9, 2026

Type Package

Title Provide Summary Information About R Objects

Version 0.1.0

Description Inspired by 'S-PLUS' function objects.summary(), provides a function with the same name that returns data class, storage mode, mode, type, dimension, and size information for R objects in the specified environment. Various filtering and sorting options are also proposed.

License GPL-3

URL https://zivankaraman.github.io/osum/, https://github.com/zivankaraman/osum

BugReports https://github.com/zivankaraman/osum/issues

Depends R (>= 3.5.0)

Imports utils

Suggests foreign, knitr, rmarkdown

VignetteBuilder knitr

Language en-US

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Zivan Karaman [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-8933-4589>)

Maintainer Zivan Karaman <zivan.karaman@gmail.com>

Repository CRAN

Date/Publication 2024-08-26 17:10:06 UTC

Contents

`objects.summary` *Summary Information About R Objects*

Description

Returns data class, storage mode, mode, typeof, dimensions, and size information for R objects from the specified environment. When invoked with no argument at the top-level prompt, `objects.summary` shows what data sets and functions a user has defined in the current session. When invoked with no argument inside a function, `objects.summary` returns the information for the function's local variables: this can be useful in conjunction with `browser`.

Usage

```
objects.summary(where, all.objects = FALSE, pattern, names. = NULL,
   what = getOption("osum.information", default = c("data.class", "storage.mode",
        "mode", "typeof", "extent", "object.size")),
    all.classes = FALSE, data.class. = NULL, storage.mode. = NULL, mode. = NULL,
    typeof. = NULL, filter., order., reverse = FALSE)
```

Arguments

`where`	which environment to use in listing the available objects. Defaults to the *current* environment. This argument can specify the environment in any form; see the 'Details' section.
`all.objects`	a logical value. If `TRUE`, information for all objects is returned. If `FALSE`, the information about objects whose names begin with a '`.`' is omitted. Default: FALSE
`pattern`	an optional regular expression. Only names matching `pattern` are returned. `glob2rx` can be used to convert wildcard patterns to regular expressions.
`names.`	an optional character vector naming objects to summarize. Default: NULL
`what`	character vector specifying what information to return. This can be any subset of `c("data.class", "storage.mode", "mode", "typeof", "extent", "object.size")`, in any order. The default is to return all six types of information, in the order shown. `what` is subject to partial matching, that is, only enough initial letters of each string element are needed to guarantee unique recognition.
`all.classes`	logical flag specifying whether the entire class vector of an object or just the first element should be used, both in selection based on argument `data.class` and in the returned summary. This has bearing only on objects with a class attribute. By default only the first class element is used.

data.class.	character vector of data classes (see data.class). Selects objects belonging to one of the named data classes. If all.classes=TRUE, each element of an object's class attribute is considered, not just the first.
storage.mode.	character vector of storage modes (see storage.mode). Selects objects with one of the named storage modes.
mode.	character vector of modes (see mode). Selects objects with one of the named modes.
typeof.	character vector of types (see typeof). Selects objects with one of the named types.
filter.	logical expression indicating elements (rows) to keep: missing values are taken as false. Note that the expression will be evaluated in the data frame with object attributes, so columns should be referred to (by name) as variables in the expression (see the examples). This argument is crafted after the select argument of the base subset function.
order.	expression involving (unquoted) attributes names, controlling the sort order of the object entries (printed as rows) in the summary. For example, order=object.size means sort the objects on the increasing values of the object.size component of the summary. order=c(data.class, -object.size) means sort the objects alphabetically by data.class, and then the decreasing values of the object.size. If order is omitted, the entries are sorted alphabetically by object name. This argument is crafted after the arguments to the base order function. The attribute names to be used in filter and order expressions must be fully specified (no partial matching possible).
reverse	logical flag: if TRUE, the final sort order is reversed, even if this order depends on object names (this is different from the original S-PLUS function). Default: FALSE

Details

The where argument can specify the environment from which object names are taken in one of several forms: as an integer (the position in the search list); as the character string name of an element in the search list; or as an explicit environment (including using sys.frame to access the currently active function calls). By default, the environment of the call to objects.summary is used.

Unless an explicit environment is provided, the where argument should designate an element of the search list. However, if it is a character of the form "package:pkg_name" and if the package named "pkg_name" is installed, it is silently loaded, its objects retrieved, and then it is unloaded when the function exits. Depending on the time it takes to load the package, the execution might be slower than getting the information about an attached package.

It is possible to use the attributes that are not returned (not listed in what) in the filter and order expressions.

Value

An object of (S3) class "objects.summary", which inherits from class "data.frame". Its components (printed as columns) are those specified in argument what. Each component contains one type of information for all selected objects. They are at most the following:

data.class	a factor (if all.classes=FALSE), or a list of character vectors (if all.classes=TRUE) containing the data class information. This is defined as in the function data.class, with the exception that when all.classes=TRUE, the summary will contain the entire class attribute for each object which has one, whereas function data.class returns only the first element of this vector.
storage.mode	a factor giving the storage mode information, as returned by function storage.mode.
mode	a factor giving the mode information, as returned by function mode.
typeof	a factor giving the R internal type or storage mode information, as returned by function typeof.
extent	a list, each of whose components is a numeric vector giving the dimension of an object, or its length if it is dimensionless.
object.size	a numeric vector giving the object sizes in bytes, as returned by function object.size.

The purpose of the dedicated class objects.summary is only to provide customized print and summary methods.

References

TIBCO Spotfire S+® 8.2 Function Guide, November 2010, TIBCO Software Inc.

See Also

glob2rx for converting wildcard patterns to regular expressions; ls.str for a long listing based on str; apropos (or find) for finding objects in the whole search path; grep for more details on 'regular expressions'; subset for filtering; order for sorting; class, data.class, methods, etc., for object-oriented programming.

Examples

```
.Ob <- 1
a <- letters[1:5]
x <- rnorm(20)
i <- 1:10
l <- list(a = a, i = i, x = x)
df <- iris
arr <- iris3
myfunc <- function() {ls()}
objects.summary()
objects.summary(pattern = "O")
objects.summary(pattern = "O", all.objects = TRUE)

objects.summary(mode = "function")
objects.summary("package:grDevices", filter = mode != "function")
objects.summary("package:datasets", all.classes = TRUE,
    filter = sapply(data.class, length) > 1)
# shows an empty list because inside myfunc no variables are defined
myfunc <- function() {objects.summary()}
myfunc()

# define a local variable inside myfunc
```

```
myfunc <- function() {y <- 1; objects.summary()}
myfunc()                # shows "y"
```

`osum.options` *Options Settings for Package* `osum`

Description

This function enables users to customize and review specific *options* for the `osum` package.

Usage

```
osum.options(
  osum.data.class.width,
  osum.format.extent,
  osum.information,
  osum.max.rows
)
```

Arguments

`osum.data.class.width`	integer indicating the width of the `data.class` field when it is a list (when `objects.summary` was called with `all.classes = TRUE`).
`osum.format.extent`	logical indicating whether the `extent` field should be formatted as product (d1 x d2) or left as list (d1, d2). Default: TRUE
`osum.information`	character vector specifying what information to return by default. This can be any subset of `c("data.class", "storage.mode", "mode", "typeof", "extent", "object.size")`, in any order. The default is to return all six types of information, in the order shown. `osum.information` is subject to partial matching, that is, only enough initial letters of each string element are needed to guarantee unique recognition.
`osum.max.rows`	integer, maximal number of rows to print.

Details

Invoking `osum.options()` with no arguments returns a list with the current values of the `osum` package options. To access the value of a single option, one can pass a character string with its name name as argument, e.g. `osum.options("osum.max.rows")`, which will return a named *list* of length one with the option's value.

Value

For `osum.options()`, a list of all the `osum` package options sorted by name. For `osum.options(name)`, a list of length one containing the set value, or `NULL` if it is unset. For uses setting one or more options, a list with the previous values of the options changed (returned invisibly).

See Also

options for global *options*, print.objects.summary and summary.objects.summary for using specific osum package options.

Examples

```
old_opt <- osum.options(osum.data.class.width = 12, osum.max.rows = 25)
cat("current values of all 'osum' options:", sep = "\n")
print(osum.options())
cat("previous values of the changed 'osum' options:", sep = "\n")
print(old_opt)
```

---

print.objects.summary *Print Method for* objects.summary *Objects*

Description

Print an object of class objects.summary with some specific formatting options before using print.data.frame.

Usage

```
## S3 method for class 'objects.summary'
print(
  x,
  ...,
  data.class.width = getOption("osum.data.class.width", default = NULL),
  format.extent = getOption("osum.format.extent", default = TRUE),
  max.rows = getOption("osum.max.rows", default = NULL)
)
```

Arguments

x	object of class objects.summary
...	further arguments to be passed down to print.data.frame (should not include max)
data.class.width	integer indicating the width of the data.class field when it is a list (when objects.summary was called with all.classes = TRUE). Default: getOption("osum.data.class.width", default = NULL)
format.extent	logical indicating whether the extent field should be formatted as product (d1 x d2) or left as list (d1, d2). Default: getOption("osum.format.extent", default = TRUE)
max.rows	integer, maximal number of rows to print. Default: getOption("osum.max.rows", default = NULL)

Details

max.rows computes an adequate value to be passed as max argument to print.data.frame. By default, when NULL, getOption("max.print") is used by print.data.frame.

Value

No return value, called for side effects.

See Also

objects.summary, print.data.frame

Examples

```
print(objects.summary("package:datasets", all.classes = FALSE),
    format.extent = FALSE, max.rows = 6)
print(objects.summary("package:datasets", all.classes = TRUE, data.class = "array"),
    data.class.width = 22, format.extent = TRUE, max.rows = 6)
```

---

summary.objects.summary

Summary Method for objects.summary *Objects*

Description

Summarize an object of class objects.summary with some specific formatting options before using summary.data.frame.

Usage

```
## S3 method for class 'objects.summary'
summary(
  object,
  ...,
  data.class.width = getOption("osum.data.class.width", default = NULL),
  format.extent = getOption("osum.format.extent", default = TRUE)
)
```

Arguments

object	object of class objects.summary
...	further arguments to be passed down to summary.data.frame
data.class.width	integer indicating the width of the data.class field when it is a list (when objects.summary was called with all.classes = TRUE). Default: getOption("osum.data.class.width", default = NULL)
format.extent	logical indicating whether the extent field should be formatted as product (d1 x d2) or left as list (d1, d2). Default: getOption("osum.format.extent", default = TRUE)

Value

A matrix of class "table", obtained by applying summary to each column of the object (after applying the specific formatting according to the arguments' values) and collating the results.

See Also

objects.summary, summary.data.frame, quantile

Examples

```
os <- objects.summary("package:datasets")
print(summary(os, format.extent = FALSE, maxsum = 10, quantile.type = 7))
print(summary(os, format.extent = TRUE, maxsum = 12, quantile.type = 1))
```

Index