

Package ‘sos’

February 14, 2010

Title sos

Version 1.2-5

Author Spencer Graves, Sundar Dorai-Raj, and Romain Francois

Maintainer Spencer Graves <spencer.graves@prodsyse.com>

Description R related Search Engines

License GPL (>= 2)

Depends brew

Suggests RODBC, WriteXLS

Repository CRAN

Repository/R-Forge/Project rsitesearch

Repository/R-Forge/Revision 167

Date/Publication 2010-02-14 19:43:43

R topics documented:

findFn	2
grepFn	4
hits	5
installPackages	6
PackageSum2	7
PackageSummary	9
print.findFn	10
sortFindFn	11
summary.findFn	12
unionFindFn	14
writeFindFn2xls	15
Index	17

findFn

*Search Help Pages***Description**

Returns a `data.frame` from `RSiteSearch(string, "function")` which can be sorted and subsetted by user specifications and viewed in an HTML table.

Usage

```
findFn(string, maxPages = 20, sortby = NULL, verbose = 1, ...)
##???string(maxPages)
```

Arguments

<code>string</code>	A character string. See <code>RSiteSearch</code> .
<code>maxPages</code>	The maximum number of pages to download assuming 20 links per page.
<code>sortby</code>	a character vector specifying how the <code>data.frame</code> returned should be sorted. Default = <code>c('Count', 'MaxScore', 'TotalScore', 'Package', 'Score', 'Function')</code> to sort descending on numerics and ascending on alphanumerics. Specifying <code>sortby = c('c', 't', 'm')</code> is equivalent to <code>c('Count', 'TotalScore', 'MaxScore', 'Package', 'Score', 'Function')</code> .
<code>verbose</code>	an integer: if 0, no output is printed to the console. The default 1 displays an initial line with the number of pages to be retrieved and the number of matches obtained; if the number of matches to be downloaded is less, this also is displayed on the initial line. This is followed by a second line counting the pages downloaded. If greater than 1, additional information is provided on the download process.
<code>...</code>	ignored

Details

`findFn` searches the help pages of packages covered by the `RSiteSearch` archives. To restrict the search to only packages installed locally, use `help.search`.

1. Access the `RSiteSearch` engine with `string`, restricting to "functions", storing `Score`, `Package`, `Function`, `Date`, `Description`, and `Link` in a `data.frame`.
2. Compute `Count`, `MaxScore` and `TotalScore` for each `Package` accessed. Combine them in a matrix `PackageSummary`.
3. Sort `PackageSummary` in the order defined by the occurrence of `c('Count', 'MaxScore', 'TotalScore', 'Package')` in `sortby`.
4. Merge `PackageSummary` with the `data.frame` of search matches.
5. Sort the combined `data.frame` as defined by `sort..`
6. Make the result have class `c("findFn", "data.frame")` and add attributes `matches`, `PackageSummary`, `string`, and `call`.
7. Done.

Value

an object of class `c('findFn', 'data.frame')` with columns and attributes as follows:

Columns	<ul style="list-style-type: none"> • <code>Count</code> Total number of matches downloaded in this package • <code>MaxScore</code> maximum of the <code>Score</code> over all help pages selected within each <code>Package</code>. See <code>Score</code> below or the Namazu website (link below) for more information on how the score is determined. • <code>TotalScore</code> sum of the <code>Score</code> over all help pages selected within each <code>Package</code>. See <code>Score</code> below or the Namazu website (link below) for more information on how the score is determined. • <code>Package Name</code> of the package containing a help page meeting the search criteria • <code>Function Name</code> of the help page found that meets the indicated search criterion. • <code>Date</code> Date of the help page • <code>Score</code> Score returned by <code>RSiteSearch</code>, discussed in the Namazu website (link below). • <code>DescriptionTitle</code> of the help page • <code>Link</code> Universal Resource Locator (URL) for the help page
Attributes	<ul style="list-style-type: none"> • <code>matches</code> an integer = total number of matches found by the search • <code>PackageSummary</code> a <code>data.frame</code> with one row for each package and columns <code>Package</code>, <code>Count</code>, <code>MaxScore</code>, <code>TotalScore</code>, and <code>Date</code>, sorted as in the <code>sort</code> argument. • <code>string</code> the <code>string</code> argument in the <code>call</code>. • <code>call</code> the matched call

Author(s)

Spencer Graves, Sundar Dorai-Raj, Romain Francois. Duncan Duncan Murdoch suggested the "???" alias for "findFn" and contributed the code for it.

Special thanks to Jonathan Baron and Andy Liaw. Baron maintains the `RSiteSearch` data base. Liaw and Baron created the `RSiteSearch` function in the `utils` package.

References

<http://www.namazu.org/doc/tips.html.en#weight> - reference on determining `Score`

See Also

`help.search` to search only installed packages. `RSiteSearch`, `download.file` <http://finzi.psych.upenn.edu/search.html> for a web interface to this same search capability with more general options. `findFn` searches only "Target: Functions" from that site, ignoring the R-help archives.

<http://www.r-project.org/search.html> for a list of alternative R search capabilities, each of which may be best for different types of inquiries.

`findFunction` for a completely different function with a similar name.

Examples

```

z <- findFn("spline", maxPages = 2)
# alternative
zq <- ???spline(2)

all.equal(z, zq)

# To search for 2 terms, not necessarily together:
RSS <- findFn('RSiteSearch function', 1)
matches(RSS)
# To search for an exact string, use braces:
RSS. <- findFn('{RSiteSearch function}', 1)
matches(RSS.) # list(nrow = 0, matches = 0)

# example in which resulting page has some unicode characters
Lambert <- findFn("Lambert")
Lambert

```

grepFn

Match pattern in a column of a matrix or data.frame

Description

Search for pattern in a column of a matrix or data.frame using grep. If value = TRUE (the default), return the selected subset of x.

Usage

```

grepFn(pattern, x, column='Function', ignore.case=FALSE,
        perl=FALSE, value=TRUE, fixed=FALSE,
        useBytes=FALSE, invert=FALSE)

```

Arguments

x a matrix or data.frame containing a column named column.

pattern, ignore.case, perl, fixed, useBytes, invert
as for grep

column character string giving the column of x in which to search for pattern.

value logical: If TRUE, return the selected subset of x. If FALSE, return the row numbers returned by grep.

Details

1. `g <- grep(pattern, x[, column])`
2. `if(value)return(x[g,]) else return(g)`

Value

If `(value)` return an object of the same class as `x` containing those rows of `x` with `x[, column]` matching `pattern`.

Else, return an integer vector identifying the rows of `x` with `x[, column]` matching `pattern`.

Author(s)

Spencer Graves, Sundar Dorai-Raj

See Also

[findFn](#) [grep](#)

Examples

```
z <- cbind(a=1:2, Function=c('s', 'spline'))
z. <- grepFn("spline", z)
```

```
all.equal(z., z[2,,drop=FALSE])
```

hits

matches attribute of a findFn object

Description

Returns the `matches` attribute of a `findFn` object. For the output of `findFn`, this is the number of matches for the search term. For a `findFn` object returned by `unionFindFn` or `intersectFindFn`, this is a numeric vector if the `matches` attributes of the arguments to `unionFindFn` or `intersectFindFn`.

Usage

```
matches(x)
hits(x)
```

Arguments

`x` object of class `findFn`.

Details

nrow(x) attr(x, 'matches')

Value

a list with components `nrows` and `matches`

Author(s)

Spencer Graves

See Also

[findFn](#) [unionFindFn](#) [intersectFindFn](#)

Examples

```
des1 <- findFn('differential equations', 1)
des1. <- matches(des1)
des. <- list(nrow=nrow(des1), matches=attr(des1, 'matches'))

all.equal(des1., des.)
```

`installPackages` *install packages with minimum count*

Description

`install.packages` identified in a `findFn` object with `Count` at least `minCount`.

Usage

```
installPackages(x, minCount=sqrt(x[1, 'Count']), ...)
```

Arguments

<code>x</code>	either a character vector to be passed to <code>install.packages</code> or a <code>findFn</code> object
<code>minCount</code>	Install all packages identified in <code>x[['Packages']]</code> with <code>x[['Count']]</code> exceeding <code>minCount</code> .
<code>...</code>	optional arguments passed to <code>install.packages</code>

Details

This is useful to prepare for `PackageSum2`.

Value

none

Author(s)

Spencer Graves

See Also

[install.packages PackageSum2](#)

Examples

```
spl <- findFn("spline", maxPages = 2)
# check the code but do not install anything:
installPackages(spl, minCount=spl[1, 'Count']+1)

# install ALL packages found
## Not run:
installPackages(spl, 1)

## End(Not run)
```

PackageSum2

Add Info from Installed Packages to PackageSummary

Description

Add information on installed packages to the PackageSummary of a findFn object.

Usage

```
PackageSum2(x,
            fields=c("Title", "Version", "Author", "Maintainer", "Packaged",
                    'helpPages', 'vignette'), lib.loc=NULL, ...)
## S3 method for class 'findFn':
PackageSum2(x,
            fields=c("Title", "Version", "Author", "Maintainer", "Packaged",
                    'helpPages', 'vignette'), lib.loc=NULL, ...)
## S3 method for class 'data.frame':
PackageSum2(x,
            fields=c("Title", "Version", "Author", "Maintainer", "Packaged",
                    'helpPages', 'vignette'), lib.loc=NULL, ...)
## S3 method for class 'list':
PackageSum2(x,
            fields=c("Title", "Version", "Author", "Maintainer", "Packaged",
                    'helpPages', 'vignette'), lib.loc=NULL, ...)
```

Arguments

<code>x</code>	a data.frame with columns <code>Package</code> and <code>Score</code> .
<code>fields</code>	character vector of names of columns to add to <code>x</code> . The function first looks in the components of <code>packageDescription(x\$Package[i])</code> . 'vignette' is obtained via the function of that name. Component 'Packaged' receives special treatment. If present, only the portion preceding ';' will be retained. This seems to be a time stamp automatically generated by something like <code>R CMD build</code> . It is absent for packages automatically loaded when R is started. In such cases, the third component of <code>strsplit(packageDescription(x\$Package[i])\$Built, ..., ';')</code> will be stored as 'Packaged'. This seems to be a time stamp automatically generated by something like <code>R CMD INSTALL --build</code> .
<code>lib.loc</code>	an optional <code>lib.loc</code> argument passed to <code>packageDescription</code> .
<code>...</code>	additional arguments (currently unused)

Details

With an object of class `findFn`, extract the `PackageSummary` attribute and pass it to the `data.frame` method.

With an object of class `list`, extract the `PackageSummary` component and pass it to the `data.frame` method.

For a `data.frame` that is not an `findFn` object, add other columns from attributes of `packageDescription` for installed packages named in the column `Package`. Also, for any packages that are installed, replace the `Date` with the `Packaged` date. The `Date` in Baron's `RSiteSearch` database is the date of acquisition, which will typically be more recent than the `Packaged` date provided the locally installed package has the same version as that in Baron's database. To get the best information from `PackageSum2`, it is wise to first run both [installPackages](#) to ensure that the packages of greatest interest are installed locally and `update.packages()` to make sure you have the latest versions installed locally. Similarly, if `PackageSum2` does not contain complete interest on a package of interest, this can be fixed by installing the package and rerunning `PackageSum2`.

Value

a data.frame with additional `fields` columns appended to a package summary data.frame.

Author(s)

Spencer Graves

See Also

[findFn](#) [PackageSummary](#) [installPackages](#)

Examples

```
##
## data.frame method
##
tstdf <- data.frame(Package=c('grid', 'base'), stringsAsFactors=FALSE)
tst2 <- PackageSum2(tstdf)

##
## list method
##
tstList <- list(PackageSummary=tstdf)

all.equal(tst2, PackageSum2(tstList))

##
## findFn method
##
tst.findFn <- data.frame(Package=c('grid', 'base')[c(1,1,2)], Score=2:4,
                        Date=LETTERS[1:3], stringsAsFactors=FALSE)
attr(tst.findFn, 'PackageSummary') <- PackageSummary(tst.findFn)
class(tst.findFn) <- c('findFn', 'data.frame')
tst2. <- PackageSum2(tst.findFn)

all.equal(tst2[, 1:7], tst2.[names(tst2)][, 1:7])
```

PackageSummary

Summarize findFn Results by Package

Description

Returns a matrix with one row for each package and columns `Count` = number of rows in the search results for that package, `maxScore` and `totalScore` = max and total score for help pages found from that package.

Usage

```
PackageSummary(x, sortBy=NULL)
```

Arguments

`x` a data.frame with columns `Package`, `Score`, and `Date`.

`sortBy` a character vector specifying how the data.frame returned should be sorted. Default = `c('Count', 'MaxScore', 'TotalScore', 'Package')` to sort descending on numerics and ascending on alphanumerics. Specifying `sortBy = c('c', 't',`

'm') is equivalent to c('Count', 'TotalScore', 'MaxScore', 'Package', 'Score', 'Function').

Components of `sortBy` must match either this list or c('Score', 'Function', 'Date', 'Description', 'Link'). Any on this latter list are ignored without a warning. This allows the same `sortBy` used for `findFn` to be used here.

Value

a data.frame with one row for each package and columns `Package`, `Count`, `MaxScore`, `TotalScore`, and `Date`, sorted as specified by `sort..`

Author(s)

Spencer Graves

See Also

[RSiteSearch](#), [findFn](#)

Examples

```
tstdf <- data.frame(Package=letters[c(1,1,2)], Score=2:4,
                  Date=LETTERS[1:3], stringsAsFactors=FALSE)
tstSum <- PackageSummary(tstdf)
# The answer:
tstSm <- data.frame(Package=letters[1:2], Count=c(a=2, b=1),
                  MaxScore=c(3, 4), TotalScore=c(5, 4),
                  Date=LETTERS[c(1, 3)], stringsAsFactors=FALSE)
row.names(tstSm) <- 1:2

all.equal(tstSum, tstSm)
```

`print.findFn` *print a findFn object*

Description

Print a `findFn` object to a file and pass it to a web browser

Usage

```
## S3 method for class 'findFn':
print(x, where, title, openBrowser = TRUE,
      template, ...)
```

Arguments

<code>x</code>	An object of class <code>findFn</code>
<code>where</code>	a character vector interpreted as follows: If <code>length(where)==1</code> , it must be either <code>'HTML'</code> or <code>'console'</code> or the name of a column of <code>x</code> or the name of a file to hold the file created to be displayed in a web browser. If <code>length(where)>1</code> , it must be the names of columns of <code>x</code> to be displayed on the console. If <code>where</code> is a vector of names of columns of <code>x</code> , those columns will be printed to the console, and there will be no display in a web browser. If <code>where == 'console'</code> , the following columns of <code>x</code> are displayed: <code>c('Count', 'Package', 'Function', 'Score', 'Date')</code> .
<code>title</code>	An optional title to give the HTML file. Default is to use the original query string.
<code>openBrowser</code>	logical; if <code>TRUE</code> and <code>where</code> is missing or <code>'HTML'</code> , launch default browser after building table
<code>template</code>	Template file used by <code>brew</code>
<code>...</code>	ignored

Value

The full path and name of the file created is returned invisibly.

Author(s)

Sundar Dorai-Raj, Spencer Graves, Romain Francois, Uwe Ligges

See Also

[findFn](#), [RSiteSearch](#), [browseURL](#), [brew](#)

Examples

```
splineSearch <- findFn("spline", maxPages = 2)
print(splineSearch, 'console')
splineSearch # all columns in a browser
```

sortFindFn

Sort a findFn Object

Description

Sort a `data.frame` as a `findFn` object.

Usage

```
sortFindFn(x, sortby=NULL)
```

Arguments

`x` a `data.frame` to sort and convert to an object of class `findFn` (if it does not already have this class).

`sortby` sort information as for function `findFn`.

Details

1. `pkgSum <- PackageSummary(x, sortby)`
2. Order `x` as required for `findFn`
3. `class = c("findFn", "data.frame")`

Value

An object of class `c('findFn', 'data.frame')` with a "PackageSummary" attribute.

Author(s)

Spencer Graves

See Also

[findFn sort order](#)

Examples

```
tstdf <- data.frame(Package=letters[c(1,1,2)],
                   Function=c('a1', 'a2', 'b3'), Score=2:4,
                   Date=11:13, Description=c('D1', 'D2', 'D3'),
                   Link=c('L1', 'L2', 'L3'), stringsAsFactors=FALSE)
rss <- sortFindFn(tstdf)
```

summary.findFn *Summary Method for findFn*

Description

Summary Method for objects of class `findFn`

Usage

```
## S3 method for class 'findFn':
summary(object, minPackages = 12,
         minCount = NA, ...)
```

Arguments

object	An object of class findFn
minPackages	the minimum number of packages to include in the summary. Other packages with the same count will also appear in the summary, but packages with a smaller count will not. The number of packages displayed will be less than minPackages only when there are fewer than that number of packages containing the search term in its help pages.
minCount	the minimum count for a package to display. minCount = 1 displays all packages. The default is the minimum of the input minCount and the count for package number minPackages.
...	ignored

Details

Return an object of class `c('summary.findFn', 'list')` with summary information on only packages satisfying the `minPackages` and `minCount` criteria. The `minPackages` and `minCount` components of the summary output list will be adjusted as necessary to match characteristics of object. The `print` method for a `summary.findFn` object will display the `minCount`, but `minPackages` will be a component of the returned object without being printed.

Value

An object of class `c('summary.findFn', 'list')` with the following elements:

PackageSummary	a data.frame with one row for each package and columns Package, Count, MaxScore, TotalScore, and Date. This summary is sorted per the <code>sortBy</code> argument in the call to <code>findFn</code> .
minPackages, minCount	the <code>minPackages</code> and <code>minCount</code> arguments in this call to <code>summary.findFn</code> .
matches	the total number of matches returned by <code>findFn</code> . This is an attribute of a <code>findFn</code> object; the number of rows of object will either be <code>matches</code> or <code>maxPages*matchesPerPage</code> , whichever is smaller.
nrow	the number of matches in this <code>findFn</code> object
nPackages	the number of packages in this <code>findFn</code> object
call	the matched call to <code>findFn</code> .

Author(s)

Spencer Graves

See Also

[findFn](#), [RSiteSearch](#)

Examples

```
z <- findFn("spline", maxPages = 2)
summary(z, 2)
```

unionFindFn	<i>Combine findFn Objects</i>
-------------	-------------------------------

Description

Combines to findFn objects into a new findFn object with only one row for any help page duplicated between the two. unionFindFn removes duplicate entries. intersectFindFn keeps only the duplicates.

Usage

```
unionFindFn(e1, e2, sortBy=NULL)
intersectFindFn(e1, e2, sortBy=NULL)

## S3 method for class 'findFn':
Ops(e1, e2)
# This supports "|" for "unionFindFn"
# and "&" for "intersectFindFn".
```

Arguments

e1, e2	objects of class findFn.
sortBy	Optional sortBy argument used by sortFindFn and findFn. Default is the sortBy argument in attr(e1, 'call').

Details

1. e12 <- rbind(e1, e2)
2. For any (Package, Function) appearing in both e1 and e2, the row with the largest Score is retained and the other is deleted.
3. Apply sortFindFn to the rebuild the summary and sort the result as desired.
4. attr(e12, 'matches') <- c(attr(e1, 'matches'), attr(e2, 'matches'))

Value

an object with class c('findFn', 'data.frame') as returned by sortFindFn and findFn.

Note

Binary operators '&' and '|' are implemented for the S3 class 'findFn'

Author(s)

Spencer Graves and Romain Francois

See Also

[findFn](#) [sortFindFn](#)

Examples

```
des1 <- findFn('differential equations', 1)
de1 <- findFn('differential equation', 1)
# each retrieves 1 page of 20 hits
# but not the same 20

de.s <- unionFindFn(des1, de1)
# combines the two, eliminating duplicates.

# or the sorter version:
de.s. <- des1 | de1

all.equal(de.s, de.s.)

# Keep only the common entries.
de2 <- intersectFindFn(des1, de1)
de2. <- des1 & de1

all.equal(de2, de2.)

# summary and print still work with the combined object.
summary(de.s)
de.s

summary(de2)
de2
```

writeFindFn2xls *Write a findFn object to an Excel file*

Description

Write a `findFn` object to an Excel file with sheets for [PackageSum2](#), the `findFn` table, and the `call` attribute of the `findFn` object.

Usage

```
writeFindFn2xls(x,
               file.=paste(deparse(substitute(x)), 'xls', sep='.'),
               csv, ...)
findFn2xls(x,
           file.=paste(deparse(substitute(x)), 'xls', sep='.'),
           csv, ...)
```

Arguments

<code>x</code>	An object of class <code>findFn</code>
<code>file.</code>	Name of Excel file to create. If a file of this name already exists, it will be overwritten.
<code>csv</code>	logical: if TRUE, write three *.csv files rather than one *.xls file. Default is FALSE if software is available to write a *.xls file and TRUE otherwise.
<code>...</code>	optional arguments to <code>write.csv</code> used if

Details

`findFn2xls` is an alias for `writeFindFn2xls`; both functions do the same thing.

1. Create the sheets
2. if(`csv` | !`require(RODBC)` | cannot otherwise create an Excel file) write.csv with names ending '-sum.csv' for `PackageSum2`, '.csv' for the main table, and '-call.csv' for the `call` sheet.
3. Open connection
4. Write the sheets
5. Done.

Value

The name of the file created is returned invisibly.

Author(s)

Spencer Graves with help from Dirk Eddedbuettel, Gabor Grothendiek, and Marc Schwartz.

See Also

[findFn](#), [odbcConnect](#), [sqlSave](#), [odbcClose](#)

Examples

```
splineSearch <- findFn("spline", maxPages = 1)

writeFindFn2xls(splineSearch)

findFn2xls(splineSearch, csv=TRUE)
```

Index

*Topic **methods**

print.findFn, 10
sortFindFn, 11
summary.findFn, 12

*Topic **misc**

findFn, 1
grepFn, 4
hits, 5
PackageSummary, 9
unionFindFn, 13

*Topic **models**

installPackages, 6

*Topic **package**

PackageSum2, 7

*Topic **print**

print.findFn, 10
sortFindFn, 11
summary.findFn, 12
writeFindFn2xls, 15

? (*findFn*), 1

??? (*findFn*), 1

brew, 11

browseURL, 11

download.file, 3

findFn, 1, 5, 8, 10–14, 16

findFn2xls (*writeFindFn2xls*), 15

findFunction, 3

grep, 5

grepFn, 4

help.search, 3

hits, 5

install.packages, 6

installPackages, 6, 8

intersectFindFn, 5

intersectFindFn (*unionFindFn*), 13

matches (*hits*), 5

odbcClose, 16

odbcConnect, 16

Ops.findFn (*unionFindFn*), 13

order, 12

PackageSum2, 6, 7, 15

PackageSummary, 8, 9

print.findFn, 10

print.summary.findFn
(*summary.findFn*), 12

RSiteSearch, 1–3, 10, 11, 13

sort, 12

sortFindFn, 11, 14

sqlSave, 16

summary.findFn, 12

unionFindFn, 5, 13

writeFindFn2xls, 15