

# Package ‘RSiteSearch’

January 2, 2012

**Title** RSiteSearch

**Version** 1.0-7

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

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

**Description** Alternative interfaces to RSiteSearch; deprecated: use 'sos'

**License** GPL

**Depends** brew, sos

**Repository** CRAN

**Repository/R-Forge/Project** rsitesearch

**Repository/R-Forge/Revision** 163

**Date/Publication** 2010-02-14 14:44:18

## R topics documented:

hits . . . . .	2
HTML . . . . .	3
PackageSum2 . . . . .	4
PackageSummary . . . . .	6
RSiteSearch.function . . . . .	7
sortRSiteSearch . . . . .	9
summary.RSiteSearch . . . . .	10
unionRSiteSearch . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

**hits***hits attribute of an RSiteSearch object*

---

**Description**

Returns the hits attribute of an RSiteSearch object. For the output of RSiteSearch.function, this is the number of matches for the search term. For an RSiteSearch object returned by unionRSiteSearch or intersectRSiteSearch, this is a numeric vector if the hits attributes of the arguments to unionRSiteSearch or intersectRSiteSearch.

**Usage**

```
hits(x)
```

**Arguments**

x                    object of class RSiteSearch.

**Details**

```
attr(x, 'hits')
```

**Value**

an integer vector

**Author(s)**

Spencer Graves

**See Also**

[RSiteSearch.function](#) [unionRSiteSearch](#) [intersectRSiteSearch](#)

**Examples**

```
des1 <- RSiteSearch.function('differential equations', 1)
all.equal(hits(des1), attr(des1, 'hits'))
```

**Description**

Create and view an HTML table from the result of calling [RSiteSearch.function](#).

**Usage**

```
HTML(x, ...)  
## S3 method for class 'RSiteSearch'  
HTML(x, file, title, openBrowser = TRUE,  
template, ...)
```

**Arguments**

x	An object of class RSiteSearch
file	An optional name for the HTML file to be created.
title	An optional title to give the HTML file. Default is to use the original query string.
openBrowser	logical; if TRUE then launch default browser after building table.
template	Template file used by brew
...	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**

[RSiteSearch.function](#), [RSiteSearch](#), [browseURL](#) [brew](#)

**Examples**

```
splineSearch <- RSiteSearch.function("spline", maxPages = 2)  
HTML(splineSearch )
```

**Description**

Add information on installed packages to the PackageSummary of an RSiteSearch object.

**Usage**

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

**Arguments**

<code>x</code>	a data.frame with columns Package and Score.
<code>fields</code>	character vector of names to look for in components of packageDescription(x\$Package[i]) and return as additional columns of x if found. 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 R CMD build. It is absent for packages automatically loaded when R is started. In such cases, the third component of strsplit(packageDescription( x\$Package[i])\$Built, ..., ';') will be stored as 'Packaged'. This seems to be a time stamp automatically generated by something like R CMD INSTALL --build.
<code>lib.loc</code>	an optional lib.loc argument passed to packageDescription.
<code>...</code>	additional arguments (currently unused)

**Details**

With an object of class RSiteSearch, 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 `RSiteSearch` object, add other columns from attributes of `packageDescription` for installed packages named in the column `Package`.

### Value

a `data.frame` with additional fields columns appended to `x`.

### Author(s)

Spencer Graves

### See Also

[RSiteSearch.function PackageSummary](#)

### Examples

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

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

all.equal(tst2, PackageSum2(tstList))

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

# row names are different from tst2, so fix ...
row.names(tst2.) <- row.names(tst2)

all.equal(tst2, tst2.[names(tst2)])
```

---

`PackageSummary`*Summarize RSiteSearch 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, sort.=NULL)
```

### Arguments

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

`sort.` 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 `sort. = c('c', 't', 'm')` is equivalent to `c('Count', 'TotalScore', 'MaxScore', 'Package', 'Score', 'Function')`. Components of `sort.` 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 `sort.` used for `RSiteSearch.function` 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](#), [RSiteSearch.function](#)

### 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)
```

---

RSiteSearch.function *Fast RSiteSearch*


---

## 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

```
RSiteSearch.function(string, maxPages = 10, sort.=NULL,
  quiet = FALSE, ...)
```

## 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>sort.</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>sort. = c('c', 't', 'm')</code> is equivalent to <code>c('Count', 'TotalScore', 'MaxScore', 'Package', 'Score', 'Function')</code> .
<code>quiet</code>	logical; if TRUE no output is printed to the console. The default FALSE displays a progress report with each page of matches.
<code>...</code>	ignored

## Details

`RSiteSearch.function` 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 `sort..`
4. Merge `PackageSummary` with the `data.frame` of search hits.
5. Sort the combined `data.frame` as defined by `sort..`
6. Make the result have class `c("RSiteSearch", "data.frame")` and add attributes `hits`, `summary`, and `call`.
7. Done.

**Value**

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

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

**Author(s)**

Spencer Graves and Sundar Dorai-Raj

**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. This function 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.

**Examples**

```
z <- RSiteSearch.function("spline", maxPages = 2)
str(z)
# To search for 2 terms, not necessarily together:
RSS <- RSiteSearch.function('RSiteSearch function', 1)
str(RSS)
# To search for an exact string, use braces:
RSS. <- RSiteSearch.function('{RSiteSearch function}', 1)
```

---

sortRSiteSearch	<i>Sort an RSiteSearch Object</i>
-----------------	-----------------------------------

---

**Description**

Sort a data.frame as an RSiteSearch object.

**Usage**

```
sortRSiteSearch(x, sort.=NULL)
```

**Arguments**

x	a data.frame to sort and convert to an object of class RSiteSearch (if it does not already have this class).
sort.	sort information as for function RSiteSearch.function.

**Details**

1. pkgSum <- PackageSummary(x, sort.)
2. Order x as required for RSiteSearch.function
3. class = c("RSiteSearch", "data.frame")

**Value**

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

**Author(s)**

Spencer Graves

**See Also**

[RSiteSearch.function 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 <- sortRSiteSearch(tstdf)
```

---

summary.RSiteSearch    *Summary Method for RSiteSearch*

---

**Description**

Summary Method for RSiteSearch

**Usage**

```
## S3 method for class 'RSiteSearch'
summary(object, threshold = 1, ...)
```

**Arguments**

object	An object of class RSiteSearch
threshold	The minimum number of hits for a package to print. Default of 1 shows all packages.
...	ignored

**Details**

1. Identify all packages with at least threshold hits.
2. Return an object of class c('summary.RSiteSearch', 'list') with summary information.

**Value**

An object of class c('summary.RSiteSearch', '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 sort. argument in the call to RSiteSearch.function.
threshold	the threshold argument in this call to summary.RSiteSearch.
hits	the total number of hits returned by RSiteSearch. This is an attribute of an RSiteSearch object; the number of rows of object will either be hits or maxPages*matchesPerPage, whichever is smaller.
nrow	the number if hits actually returned by RSiteSearch.function.
call	the matched call to RSiteSearch.function.

**Author(s)**

Spencer Graves

**See Also**[RSiteSearch.function](#), [RSiteSearch](#)**Examples**

```
z <- RSiteSearch.function("spline", maxPages = 2)
summary(z, 2)
```

---

unionRSiteSearch	<i>Combine RSiteSearch Objects</i>
------------------	------------------------------------

---

**Description**

Combines to RSiteSearch objects into a new RSiteSearch object with only one row for any help page duplicated between the two. unionRSiteSearch removed the duplicate entries. intersectRSiteSearch keeps only the duplicates.

**Usage**

```
unionRSiteSearch(x, y, sort.=NULL)
intersectRSiteSearch(x, y, sort.=NULL)

## S3 method for class 'RSiteSearch'
Ops(e1,e2)
# This supports "|" for "unionRSiteSearch"
# and "&" for "intersectRSiteSearch".
```

**Arguments**

x, y	objects of class RSiteSearch.
sort.	Optional sort. argument used by sortRSiteSearch and RSiteSearch.function. Default is the sort. argument in attr(x, 'call').
e1, e2	objects of class RSiteSearch.

**Details**

1. xy <- rbind(x, y)
2. For any (Package, Function) appearing in both x and y, the row with the largest Score is retained and the other is deleted.
3. Apply sortRSiteSearch to the rebuild the summary and sort the result as desired.
4. attr(xy, 'hits') <- c(attr(x, 'hits'), attr(y, 'hits'))

**Value**

an object with class `c('RSiteSearch', 'data.frame')` as returned by `sortRSiteSearch` and `RSiteSearch.function`.

**Note**

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

**Author(s)**

Spencer Graves and Romain Francois

**See Also**

[RSiteSearch.function](#) [sortRSiteSearch](#)

**Examples**

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

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

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

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

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

summary(de2)
HTML(de2)
```

# Index

## \*Topic **methods**

HTML, [3](#)  
sortRSiteSearch, [9](#)  
summary.RSiteSearch, [10](#)

## \*Topic **misc**

hits, [2](#)  
PackageSummary, [6](#)  
RSiteSearch.function, [7](#)  
unionRSiteSearch, [11](#)

## \*Topic **package**

PackageSum2, [4](#)

## \*Topic **print**

HTML, [3](#)  
sortRSiteSearch, [9](#)  
summary.RSiteSearch, [10](#)

brew, [3](#)

browseURL, [3](#)

download.file, [8](#)

help.search, [8](#)

hits, [2](#)

HTML, [3](#)

intersectRSiteSearch, [2](#)

intersectRSiteSearch  
(unionRSiteSearch), [11](#)

Ops.RSiteSearch (unionRSiteSearch), [11](#)

order, [9](#)

PackageSum2, [4](#)

PackageSummary, [5](#), [6](#)

print.summary.RSiteSearch  
(summary.RSiteSearch), [10](#)

RSiteSearch, [3](#), [6–8](#), [11](#)

RSiteSearch.function, [2](#), [3](#), [5](#), [6](#), [7](#), [9](#), [11](#), [12](#)

sort, [9](#)

sortRSiteSearch, [9](#), [12](#)

summary.RSiteSearch, [10](#)

unionRSiteSearch, [2](#), [11](#)