

Package ‘pkgDepTools’

October 18, 2022

Type Package

Title Package Dependency Tools

Version 1.62.0

Description This package provides tools for computing and analyzing dependency relationships among R packages. It provides tools for building a graph-based representation of the dependencies among all packages in a list of CRAN-style package repositories. There are also utilities for computing installation order of a given package. If the RCurl package is available, an estimate of the download size required to install a given package and its dependencies can be obtained.

License GPL-2

Depends methods, graph, RBGL

Imports graph, RBGL

Suggests Biobase, Rgraphviz, RCurl, BiocManager

LazyLoad Yes

biocViews Infrastructure, GraphAndNetwork

git_url <https://git.bioconductor.org/packages/pkgDepTools>

git_branch RELEASE_3_15

git_last_commit c448b71

git_last_commit_date 2022-04-26

Date/Publication 2022-10-18

Author Seth Falcon [aut],
Bioconductor Package Maintainer [cre]

Maintainer Bioconductor Package Maintainer <maintainer@bioconductor.org>

R topics documented:

pkgDepTools-package	2
basicInstallOrder	2

cleanPkgField	3
getDownloadSize	4
getInstallOrder	4
makeDepGraph	5
makePkgUrl	6
parseContentLength	7

Index	8
--------------	----------

pkgDepTools-package *Package Dependency Tools*

Description

This package provides tools for computing and analyzing dependency relationships among R packages. It provides tools for building a graph-based representation of the dependencies among all packages in a list of CRAN-style package repositories. There are also utilities for computing installation order of a given package. If the RCurl package is available, an estimate of the download size required to install a given package and its dependencies can be obtained.

Details

Package: pkgDepTools
 Type: Package
 License: GPL2

Author(s)

Seth Falcon Maintainer: Seth Falcon <sfalcon@fhcrc.org>

basicInstallOrder *Complete Installation Order of a Given Package*

Description

Helper function to return the complete install order of a given package.

Usage

```
basicInstallOrder(pkg, depG)
```

Arguments

pkg character string package name
depG graphNEL instance as returned by makeDepGraph

Details

Internal function.

Value

character vector of package names in a viable installation order.

Author(s)

Seth Falcon

cleanPkgField *Clean Packages Names from a DESCRIPTION File Field*

Description

Given the value from a field like 'Depends' in a package's DESCRIPTION file, return a character vector of package names with the version restrictions stripped and R removed.

Usage

cleanPkgField(val)

Arguments

val character string parsed from a package's DESCRIPTION file containing package names and possible version specifiers.

Details

Internal function

Warning

FIXME: uses a private function from tools

Author(s)

Seth Falcon

getDownloadSize	<i>Get Size in MB of a URL Using RCurl</i>
-----------------	--

Description

Returns the size in MB of the specified URL. Uses RCurl to parse the HTTP headers for Content-Length.

Usage

```
getDownloadSize(url)
```

Arguments

url character vector giving the URL. This is not vectorized.

Note

Internal function.

Author(s)

Seth Falcon

getInstallOrder	<i>List package dependencies in install order</i>
-----------------	---

Description

This function uses a dependency graph created with makeDepGraph to list all (recursive) dependencies of a given package in an order suitable for installation.

Usage

```
getInstallOrder(pkg, depG, needed.only = TRUE)
```

Arguments

pkg character string name of package
depG graphNEL instance as returned from makeDepGraph.
needed.only logical value. When TRUE, only those dependencies not currently installed are included in the list, this is the default. When FALSE the complete list of dependencies is given regardless of the set of currently installed packages.

Value

a list with components:

packages	character vector of package names in an order suitable for installation. The order is not unique. The names of the character vector are the estimated download sizes of each package.
total.size	When available, the total size in megabytes of all listed dependencies.

Author(s)

Seth Falcon

makeDepGraph	<i>Create a graph representing R package dependencies</i>
--------------	---

Description

Given a list of CRAN-style repository URLs, this function creates a graph instance representing dependencies between packages in the repositories.

Usage

```
makeDepGraph(repList, suggests.only = FALSE, type =
getOption("pkgType"), keep.builtin = FALSE, dosize = TRUE)
```

Arguments

repList	a character vector of URLs pointing to CRAN-style repositories.
suggests.only	logical value indicating whether only the "Suggests" field should be used. The default (FALSE) means that DESCRIPTION fields "Depends" and "Imports" are used. Note that "Suggests" is special in that its use is less regulated and the resulting graph is less likely to be a DAG (packages sometimes suggest each other).
type	a character vector specifying the type of packages to search for in the repositories. Must be one of "source", "win.binary", or "mac.binary".
keep.builtin	logical value indicating whether or not packages that come with a default R installation should be included in the nodes of the dependency graph. The default, FALSE, is to remove these packages from the return result.
dosize	logical value. When TRUE, the function will attempt to estimate the download size of each package (requires the RCurl package). The size of each package is added as a node attribute to the graph.

Value

A graphNEL-class instance. If the dosize argument was set to TRUE, then an estimate of the download size of each package is stored as a node attribute of the graph. A missing value is used when the download size was not able to be determined as well as when the RCurl package is not available.

Author(s)

Seth Falcon

Examples

```
## Not run:
  deps <- makeDepGraph("http://cran.fhcrc.org", type="source")

## End(Not run)
```

`makePkgUrl`*Create a URL for a Package in a CRAN-Style Repository*

Description

Given a CRAN-style package repository meta data matrix as returned by `available.packages`, return a character vector of complete URLs for each package.

Usage

```
makePkgUrl(pMat, type = getOption("pkgType"))
```

Arguments

<code>pMat</code>	matrix as returned by <code>available.packages</code>
<code>type</code>	character string, used to determine file extension. Should match the argument used to generate <code>pMat</code> .

Details

Internal function.

Author(s)

Seth Falcon

`parseContentLength` *Parse HTTP Response Header for Content Length*

Description

Given a string containing an HTTP response header, extract the "Content-Length" header and convert it to a numeric value. If no such header is found, return NA.

Usage

`parseContentLength(h)`

Arguments

`h` string containing the HTTP response header

Details

Internal function

Author(s)

Seth Falcon

Index

* **package**

pkgDepTools-package, [2](#)

* **utilities**

basicInstallOrder, [2](#)

cleanPkgField, [3](#)

getDownloadSize, [4](#)

getInstallOrder, [4](#)

makeDepGraph, [5](#)

makePkgUrl, [6](#)

parseContentLength, [7](#)

[basicInstallOrder, 2](#)

[cleanPkgField, 3](#)

[getDownloadSize, 4](#)

[getInstallOrder, 4](#)

[makeDepGraph, 5](#)

[makePkgUrl, 6](#)

[parseContentLength, 7](#)

[pkgDepTools \(pkgDepTools-package\), 2](#)

[pkgDepTools-package, 2](#)