

Package ‘miniCRAN’

July 29, 2014

License GPL-2 | GPL-3

Title Tools to create an internally consistent, mini version of CRAN with selected packages only.

LazyData true

LazyLoad true

Description Tools to create a internally consistent, mini version of CRAN with selected packages only.

Version 0.0-16

URL <https://github.com/andrie/miniCRAN>

Date 2014-07-29

Imports tools, XML

Suggests igraph, RCurl, testthat, knitr

VignetteBuilder knitr

Author Andrie de Vries [aut, cre]

Maintainer Andrie de Vries <apdevries@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-07-29 19:44:51

R topics documented:

miniCRAN-package	2
basePkgs	3
cranJuly2014	3
getCranDescription	3
makeDepGraph	4
makeRepo	5
makeRsyncInclude	6

pkgAvail	7
pkgDep	7
plot.pkgDepGraph	8
Index	11

miniCRAN-package	<i>Tools to create an internally consistent, mini version of CRAN with selected packages only.</i>
------------------	--

Description

Tools to create an internally consistent, mini version of CRAN with selected packages only.

Details

At the end of 2013, CRAN consisted of more than 5000 packages. Many organisations need to maintain a private mirror of CRAN, but with only a subset of packages that are relevant to them.

miniCRAN makes this possible by recursively reading the dependency tree for a given set of packages, then downloading only this subset.

Important functions:

- [pkgDep](#): Find (recursive) package dependencies
- [makeRepo](#) : Make repository
- [pkgAvail](#): Read local repository and determine available packages
- [makeDepGraph](#): Create graph of selected package dependencies

Author(s)

Andrie de Vries <apdevries@gmail.com>

See Also

[minicran](#)

basePkgs	<i>Returns names of base packages.</i>
----------	--

Description

Retrieves names of installed packages by calling [installed.packages](#) and returning only those packages where Priority equals "base".

Usage

```
basePkgs()
```

See Also

[pkgDeps](#)

cranJuly2014	<i>Stored version of available.packages()</i>
--------------	---

Description

Copy of the result of [available.packages](#) on July 1, 2014.

Usage

```
cranJuly2014
```

Format

```
matrix
```

getCranDescription	<i>Scrape DESCRIPTION from CRAN for each pkg.</i>
--------------------	---

Description

Scrape DESCRIPTION from CRAN for each pkg.

Usage

```
getCranDescription(pkg, repos = getOption("repos"), type = "source", path,  
  pkgs = pkgDep(pkg, repos = repos, type = type))
```

Arguments

pkg	Character vector of packages.
repos	URL(s) of the 'contrib' sections of the repositories. Passed to available.packages
type	Passed to available.packages
path	Destination download path
pkgs	Character vector of packages to download

See Also

Other miniCRAN: [pkgAvail](#); [pkgDep](#)

Examples

```
## Not run:
getCranDescription(c("igraph", "ggplot2", "XML"),
  repos=c(CRAN="http://cran.revolutionanalytics.com")
)
## End(Not run)
```

makeDepGraph

Create dependency graph from available packages.

Description

Each package is a node, and a dependency is an edge

Usage

```
makeDepGraph(pkg, availPkgs, repos = getOption("repos"), type = "source",
  path, suggests = TRUE, enhances = FALSE, includeBasePkgs = FALSE, ...)
```

Arguments

pkg	Character vector of packages.
availPkgs	Vector of available packages. Defaults to reading this list from CRAN, using available.packages
repos	URL(s) of the 'contrib' sections of the repositories. Passed to available.packages
type	Passed to available.packages
path	Destination download path
suggests	If TRUE, retrieves Suggests dependencies (non-recursively)
enhances	If TRUE, retrieves Enhances dependencies (non-recursively)
includeBasePkgs	If TRUE, include base R packages in results
...	Other arguments passed to available.packages

See Also

pkgDep plot.pkgDepGraph

Examples

```
p <- makeDepGraph(
  c("ggplot2", "forecast"),
  repos = c(CRAN="http://cran.revolutionanalytics.com"),
  type="source"
)
if(require(igraph)) plot(p)
```

```
availPkgs <- pkgAvail(
  repos = c(CRAN="http://cran.revolutionanalytics.com"),
  type="source"
)
```

```
p <- makeDepGraph(
  c("ggplot2", "forecast"),
  availPkgs = availPkgs
)
```

```
if(require(igraph)) plot(p)
```

makeRepo

Downloads packages from CRAN to specified path and creates repository or library.

Description

Given a list of packages, downloads to a specified destination folder, then creates PACKAGES file.

Usage

```
makeRepo(pkgs, path, repos = getOption("repos"), type = "source",
  Rversion = getRversion(), download = FALSE, writePACKAGES = TRUE)
```

```
getRversion()
```

```
makeLibrary(pkgs, path, type = "source")
```

Arguments

pkgs	Character vector of packages to download
type	Passed to download.packages
Rversion	String of format "<major R version>.<minor R version>", e.g. "3.2". Only used if type is not "source"
download	If TRUE downloads packages, otherwise just creates PACKAGES file
writePACKAGES	If TRUE, calls write_PACKAGES to update the repository PACKAGES file <pre># Make repo for source and win.binary makeRepo(pkgList, path=pth, repos=revolution, download=TRUE, writePACKAGES=TRUE, type="source") makeRepo(pkgList, path=pth, repos=revolution, download=TRUE, writePACKAGES=TRUE, type="win.binary") # List all files in miniCRAN list.files(pth, recursive = TRUE) # Check for available packages pkgAvail(repos=pth, type="source") pkgAvail(repos=pth, type="win.binary") # Delete temporary folder unlink(pth, recursive = TRUE)</pre>
path	Destination download path
repos	URL(s) of the 'contrib' sections of the repositories. Passed to available.packages

Details

The function `makeRepo` creates a repository, similar in structure to CRAN. It optionally updates the PACKAGES file. If done correctly, it is possible to use this folder as a repository, i.e. it will support functions like `install.packages`.

The function `makeLibrary` downloads the packages into a single folder, i.e. similar to a library on a machine.

Uses [download.packages](#) and [write_PACKAGES](#)

`makeRsyncInclude` *Create include file for use with rsync.*

Description

Create include file for use with rsync.

Usage

```
makeRsyncInclude(pkg, file = NULL, pkgs = pkgDep(pkg, ...), ...)
```

Arguments

file	Name of file where results are saved. If NULL, the function returns a vector.
pkg	Character vector of packages.
...	Other arguments passed to available.packages
pkgs	Character vector of packages to download

Examples

```
pkgs <- c("ggplot2", "plyr", "reshape2")
makeRsyncInclude(pkgs, type="source", repos=c(CRAN="http://cran.revolutionanalytics.com"))
```

pkgAvail	<i>Reads available packages from CRAN repository.</i>
----------	---

Description

This is a thin wrapper around [available.packages](#). If the argument `path` is supplied, then the function attempts to read from a local repository, otherwise attempts to read from a CRAN mirror at the `repos` url.

Usage

```
pkgAvail(repos = getOption("repos"), type = "source", ...)
```

Arguments

<code>repos</code>	URL(s) of the 'contrib' sections of the repositories. Passed to available.packages
<code>type</code>	Passed to available.packages
<code>...</code>	Other arguments passed to available.packages

See Also

Other miniCRAN: [getCranDescription](#); [pkgDep](#)

pkgDep	<i>Retrieves package dependencies.</i>
--------	--

Description

Performs recursive retrieve for Depends, Imports and LinkLibrary. Performs non-recursive retrieve for Suggests.

Usage

```
pkgDep(pkg, availPkgs, repos = getOption("repos"), type = "source",
  depends = TRUE, suggests = TRUE, enhances = FALSE, path,
  includeBasePkgs = FALSE, ...)
```

Arguments

pkg	Character vector of packages.
availPkgs	Vector of available packages. Defaults to reading this list from CRAN, using available.packages
repos	URL(s) of the 'contrib' sections of the repositories. Passed to available.packages
type	Passed to available.packages
depends	If TRUE, retrieves Depends, Imports and LinkingTo dependencies (non-recursively)
suggests	If TRUE, retrieves Suggests dependencies (non-recursively)
enhances	If TRUE, retrieves Enhances dependencies (non-recursively)
path	Destination download path
includeBasePkgs	If TRUE, include base R packages in results
...	Other arguments passed to available.packages

Details

This

See Also

[makeDepGraph](#)

Other miniCRAN: [getCranDescription](#); [pkgAvail](#)

Examples

```
pkgDep(pkg=c("ggplot2", "plyr", "reshape2"),  
        repos=c(CRAN="http://cran.revolutionanalytics.com")  
)
```

```
pdb <- pkgAvail()  
pkgDep(pkg=c("ggplot2", "plyr", "reshape2"), pdb)
```

plot.pkgDepGraph

Plots a package dependency graph.

Description

Plots a package dependency graph.

Usage

```
## S3 method for class 'pkgDepGraph'
plot(x, pkgsToHighlight,
     main = "Package dependency graph", legendPosVertex = c(-1, -1),
     legendPosEdge = c(1, -1), shape = "circle", vertex.size = 8, cex = 1,
     ...)
```

Arguments

x	pkgDepGraph object
pkgsToHighlight	Optional character vector with names of package to highlight. If missing, defaults to packages used in original call to makeDepGraph
main	Title of plot
legendPosVertex	Numeric vector of length 2, indicating (x, y) position of vertex legend. Both values should be in the range [-1; 1]. If NULL, the vertex legend is not displayed.
legendPosEdge	Numeric vector of length 2, indicating (x, y) position of edge legend. Both values should be in the range [-1; 1]. If NULL, the edge legend is not displayed.
shape	Shape of edge. See igraph.plotting . Could be "none", "circle", "square", ...
vertex.size	Size of vertex shape. See igraph.plotting
cex	Vertex label size.
...	Ignored

See Also

[makeDepGraph](#)

Examples

```
tags <- "chron"

# Plot using defaults
dg <- makeDepGraph(tags, includeBasePkgs=FALSE, suggests=TRUE, enhances=TRUE)

set.seed(42);
plot(dg)

# Move edge legend to top left
set.seed(42);
plot(dg, legendPosEdge=c(-1, 1))

# Change font size and shape size
set.seed(42);
plot(dg, legendPosEdge=c(-1, 1), vertex.size=20, cex=0.5)

# Move vertex legend to top right
```

```
set.seed(42);  
plot(dg, legendPosEdge=c(-1, 1), legendPosVertex=c(1, 1), vertex.size=20, cex=0.5)
```

Index

*Topic **datasets**

cranJuly2014, 3

*Topic **package**

miniCRAN-package, 2

available.packages, 3, 4, 6–8

basePkgs, 3

cranJuly2014, 3

download.packages, 6

getCranDescription, 3, 7, 8

getRversion (makeRepo), 5

igraph.plotting, 9

installed.packages, 3

makeDepGraph, 2, 4, 9

makeLibrary (makeRepo), 5

makeRepo, 2, 5

makeRsyncInclude, 6

miniCRAN (miniCRAN-package), 2

minicran, 2

minicran (miniCRAN-package), 2

miniCRAN-package, 2

pkgAvail, 2, 4, 7, 8

pkgDep, 2, 4, 7, 7

plot.pkgDepGraph, 8

write_PACKAGES, 6