Package ‘NoviceDeveloperResources2’

February 1, 2024

Version 1.0

Date 2024-01-30

Title Further Resources to Assist Novice Developers

Maintainer Barry Zeeberg <barryz2013@gmail.com>

Depends R (>= 4.2.0)

Imports packrat, NoviceDeveloperResources

Description Assist novice developers when preparing a single package or a set of integrated packages to submit to CRAN. Provide additional resources to facilitate the automation of the following individual or batch processing: check local source packages; build local .tar.gz source files; install packages from local .tar.gz files; detect conflicts between function names in the environment. The additional resources include determining the identity and ordering of the packages to process when updating an imported package.

License GPL (>= 2)

Encoding UTF-8

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation no

Author Barry Zeeberg [aut, cre]

Repository CRAN

Date/Publication 2024-02-01 11:00:09 UTC

R topics documented:

bottomUpRecursive *bottomUpRecursive*

Description

given a list of packages, determine which packages recursively import the packages in the list

Usage

```
bottomUpRecursive(l, p0)
```

Arguments

`l`	return value of recursivePackageDependenciesWrapper()
`p0`	list of those packages whose R code has been modified by the developer

Value

returns a list of the original query packages plus the packages that directly import them

Examples

```
## Not run:
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
l<-recursivePackageDependenciesWrapper(packs)
bur<-bottomUpRecursive(l,c("iterationDriver"))

## End(Not run)
```

`bottomUpRecursiveDriver` *bottomUpRecursiveDriver*

Description

compute a list of all the packages that either directly or indirectly import the original query packages

Usage

```
bottomUpRecursiveDriver(l, p0, verbose)
```

Arguments

`l`	return value of recursivePackageDependenciesWrapper()
`p0`	list of those packages whose R code has been modified by the developer
`verbose`	if TRUE print line indicating the recursion level

Value

returns a list of all the packages that either directly or indirectly imports the original query packages

Examples

```
## Not run:
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
l<-recursivePackageDependenciesWrapper(packs)
burd<-bottomUpRecursiveDriver(l,c("iterationDriver"),TRUE)

## End(Not run)
```

`PackageDependencies` *PackageDependencies*

Description

recursively call recursivePackageDependencies2() and reversePackageDependencies() to recursively delete leaf nodes until packs has been depleted to length zero

Usage

```
PackageDependencies(packs, master, n, verbose)
```

Arguments

packs	list of package names
master	list whose componenets are lists indexed by integer recursion level the components of each recursion level are the return values of recursivePackageDependenciesWrapper() and reversePackageDependencies()
n	integer recursion level
verbose	if TRUE print line indicating the recursion level

Details

NOTE that the packages in packs do not need to be loaded or attached to the search() path

Value

returns a list whose components are lists indexed by the integer recursion level:

l	return value of recursivePackageDependenciesWrapper()
ll	return value of reversePackageDependencies()

Examples

```
## Not run:
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
master<-PackageDependencies(packs, vector("list",length(packs)),1,TRUE)

## End(Not run)
```

recursivePackageDependenciesWrapper

recursivePackageDependenciesWrapper

Description

determine which packages in packs list have length zero dependencies

Usage

```
recursivePackageDependenciesWrapper(packs)
```

Arguments

packs	list of package names

Details

the return value l is like:

$cardUtils

character(0) [i.e., cardUtils has no further dependencies within packs list]

$editDriver

[1] "cardUtils" "clickableImageMap" "heartsCIM" "logos" "probTab"

Value

returns a list of lists. Each component is indexed by a package name, and contains a list of package names that are imported by the index package name

Examples

```
## Not run:
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
l<-recursivePackageDependenciesWrapper(packs)

## End(Not run)
```

`retrieveLeafNodes` *retrieveLeafNodes*

Description

compute a list of the packages in the correct order for processing by checkBuildInstallSourcePackage()

Usage

```
retrieveLeafNodes(master)
```

Arguments

`master`	return value of sortedInputForCheckBuildInstallSourcePackageDriver()

Details

the master list may contain some packages that do not need to be processed by checkBuildInstallSourcePackage(). These are weeded out by sortedInputForcheckBuildInstallSourcePackage()

Value

returns a list of the packages in the correct order for processing by checkBuildInstallSourcePackage()

Examples

```
## Not run:
dir<-"~/inference_packages/"
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
master<-PackageDependencies(packs,vector("list",length(packs)),1,TRUE)
retrieve<-retrieveLeafNodes(master)

## End(Not run)
```

`reversePackageDependencies`

reversePackageDependencies

Description

separate the packages in packs list having length zero or non-zero dependencies

Usage

```
reversePackageDependencies(l)
```

Arguments

`l` return value of recursivePackageDependenciesWrapper()

Details

the return value ll is like:

$original [cardUtils is no longer a name of ll$original since it had length 0]

$original$editDriver

[1] "cardUtils" "clickableImageMap" "heartsCIM" "logos" "probTab"

$zeros

$zeros$cardUtils cardUtils is an element of ll$zeros since l[["cardUtils"]] has length 0

[1] "cardUtils"

Value

returns a list whose components are 2 lists:

`$original` a list whose components are lists of package names that have non-zero length import dependencies ll$original is same as l, but deleting zero-length elements i.e., leaf nodes

`$zeros` a list whose components are lists of package names that have zero length import dependencies (i.e., leaf nodes) ll$zeros zero-length elements, leaf nodes that had been deleted in ll$original

Examples

```
## Not run:
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
l<-recursivePackageDependenciesWrapper(packs)
ll<-reversePackageDependencies(l)

## End(Not run)
```

`sortedInputForCheckBuildInstallSourcePackage` *sortedInputForCheckBuildInstallSourcePackage*

Description

compute a list of packages in the correct order to input to checkBuildInstallSourcePackage()

Usage

```
sortedInputForCheckBuildInstallSourcePackage(retrieve, burd)
```

Arguments

`retrieve`	return value of retrieveLeafNodes()
`burd`	return value of bottomUpRecursiveDriver()

Value

returns a list of packages in the correct order to input to checkBuildInstallSourcePackage()

Examples

```
## Not run:
dir<-"~/inference_packages/"
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
master<-PackageDependencies(packs,vector("list",length(packs)),1,TRUE)
l<-recursivePackageDependenciesWrapper(packs)
burd<-bottomUpRecursiveDriver(l,c("iterationDriver"),TRUE)
retrieve<-retrieveLeafNodes(master)
s<-sortedInputForCheckBuildInstallSourcePackage(retrieve,burd)

## End(Not run)
```

sortedInputForCheckBuildInstallSourcePackageDriver

sortedInputForCheckBuildInstallSourcePackageDriver

Description

driver to invoke sequence of functions to retrieve the correctly ordered list of packages as input and to invoke checkBuildInstallSourcePackage()

Usage

```
sortedInputForCheckBuildInstallSourcePackageDriver(dir, packs, p0, verbose)
```

Arguments

`dir`	character string containing the path name of the directory holding the package folders
`packs`	list of package names
`p0`	list of those packages whose R code has been modified by the developer
`verbose`	if TRUE print line indicating the recursion level

Details

This driver is the single master function to run in order to invoke all of the other functions in the packages *NoviceDeveloperResources* and *NoviceDeveloperResources2*.

In the examples, I show the actual call using packages that are currently under development, so they are not yet available (I expect them to be available in mid-2024).

Value

a list whose components are the return values of checkBuildInstallSourcePackage() and conflictOfInterestRestricted()

Examples

```
## Not run:
dir<-"~/inference_packages/"
packs<-c("cardUtils","clickableImageMap","editDriver",
"heartsCIM","iterationDriver","logos","playOneTrick",
"playWholeHandDriverPassParams","probTab","relaxDriver")
l<-sortedInputForCheckBuildInstallSourcePackageDriver(dir,packs,"iterationDriver",TRUE)

## End(Not run)
```

Index