Package 'lazyData'

October 13, 2022

Type Package

Title A LazyData Facility

Version 1.1.0

Date 2016-12-05

Author Bill Venables

Maintainer Bill Venables <bill.venables@gmail.com>

Description Supplies a LazyData facility for packages which have data sets but do not provide LazyData: true. A single function is is included, requireData, which is a drop-in replacement for base::require, but carrying the additional functionality. By default, it suppresses package startup messages as well. See argument 'reallyQuitely'.

Depends R (>= 2.15.0)

Imports utils

Suggests mgcv

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-04 14:52:35

R topics documented:

`lazyData-package` *A LazyData Facility*

Description

Supplies a `LazyData` facility for packages which have data sets but do not provide `LazyData`. A single function is is included, `requireData`, which is a drop-in replacement for `base::require`, but carrying the additional functionality.

Details

Package:	lazyData
Type:	Package
Version:	1.1.0
Date:	2016-12-05
License:	GPL-2

If a package is attached with `requireData`, a check is made to see if a) the package provides data and b) if the data sets are not made visible to the user via `LazyData: true`. If both are the case, the package is attached to the search path, and in addition a second entry to the search path is made immediately behind the package containing `promises` to load the data sets should they be needed. This keeps data sets out of the global environment (unless they are modified), making them visible without occupying memory (unless they are needed).

Using `requireData` a second time on the same package has the effect of flushing any data sets brought into memory and reinstating them as promises.

Author(s)

Bill Venables

Maintainer: Bill Venables <bill.venables@gmail.com>

References

None.

See Also

`require`, `library`, `data` `delayedAssign`

Examples

```
requireData("mgcv") ## we assume has data sets but no LazyLoad
.Search()           ## show the augmented search path
```

```
## > ls("datasets:mgcv")
##[1] "columb"       "columb.polys"
```

`requireData` *Attach packages as required and expose non-LazyData data sets as promises.*

Description

This function provides `LazyData` functionality for packages which do not provide it. It acts as an enhanced substitute for the base packges `require` function.

Usage

```
requireData(package = stop("you must specify a package"),
            lib.loc = NULL, quietly = TRUE, character.only = FALSE,
            warn.conflicts = TRUE, reallyQuietly = TRUE, ...)
```

Arguments

`package`	The name of the package whose attachment to the search path is required. May me a name or a literal character string.
`lib.loc`	The path to the library holding the package. As for `require`.
`quietly`	Logical: should the stanandard loading message be suppressed? Ignored if `reallyQuiet` is `TRUE`.
`character.only`	Logical: should the `package` argument be treated as a character string even if not literal?
`warn.conflicts`	Should objects masked by the attachment of the package be flagged? As for `require`. Ignored if `reallyQuiet` is `TRUE`.
`reallyQuietly`	Logical: should the `package` be loaded using `suppressPackageStartupMessages`? If `TRUE`, the default, this will make the loading as quietly as possible, but will suppress potentially useful messages, such as masking information.
`...`	Additional arguments currently ignored.

Details

The only function this package provides, `requireData`, is a substitute for the base function `require`. If the package is not already on the search path, it attaches it. In addition, if the package a) has data sets and b) does NOT use the LazyData facility, then an additonal entry is made on the search path. This is an unlocked environment initially populated by 'promises' (using `delayedAssign`) to load a copy of the data set into memory if and when it is needed.

This is done recursively for all packages attached to the search path via dependencies.

If the package appears on the search path as `package:<pkg>` at positon `p`, then any exposed data set objects appear at postion `p+1` as `datasets:<pkg>`. The package environment is locked, but the

datasets environment is not. If a data set object is needed at any stage, it is brought silently into memory at position p+1 on the search path.

Any further call to requireData(<pkg>) will reinstate the datasets as promises, thus potentially freeing memory.

The intended effect is to make data sets more conveniently available to users, to make the use of the data function largely unnecessary, and to avoid cluttering the global environment with copies of passive data set objects.

Value

TRUE if the package was successfully attached and FALSE otherwise.

Author(s)

Bill Venables

References

Null

See Also

require, data

Examples

```
requireData("mgcv") ## we assume has data sets but no LazyLoad
.Search()           ## show augmented search path

## > ls("datasets:mgcv")
##[1] "columb"       "columb.polys"
```

Index