

DynDoc

October 5, 2010

DynDoc-class

A Class For Dynamic Documents

Description

The DynDoc class is used to represent dynamic documents and vignettes in R.

Slots

indexEntry: Object of class "character" The IndexEntry value from the document file
title: Object of class "character" The name of the document
path: Object of class "character" The path to the locally stored file
pdfPath: Object of class "character" The path to a PDF rendition of the document
depends: Object of class "character" Any package dependencies for the document
requires: Object of class "character" Any requires level dependencies for the document
suggests: Object of class "character" Any suggests level dependencies for the document
keywords: Object of class "character" Any keywords for the document
codeChunks: Object of class "chunkList" The code chunks contained in this document

Methods

show signature(object = "DynDoc"): Display information about the dynamic document
summary signature(object = "DynDoc"): A more succinct informational display
chunks signature(object = "DynDoc"): Returns the code chunks - currently in only for historical compatability with old code
codeChunks signature(object = "DynDoc"): Returns the code chunks
evalChunk signature(object = "DynDoc"): Will evaluate the R code contained in a chunk
getChunk signature(object = "DynDoc"): Retrieves a specific code chunk
getDepends signature(object = "DynDoc"): Obtain the Depends slot of the object
getKeywords signature(object = "DynDoc"): Obtain the keywords slot of the object
getRequires signature(object = "DynDoc"): A get method for the requires slot of this object

getSuggests signature(object = "DynDoc"): Obtain the suggests slot of this object
indexEntry signature(object = "DynDoc"): Obtain the indexEntry slot of this object
numChunks signature(object = "DynDoc"): Returns the number of code chunks for this document
path signature(object = "DynDoc"): Obtain the path slot of this object
pdfPath signature(object = "DynDoc"): Obtain the pdfPath slot of this object
setChunk<- signature(object = "DynDoc"): Change the code for one of the code chunks.

Author(s)

Jeff Gentry

See Also

[Sweave](#)

SweaveOptions-class

Class "SweaveOptions", a class to handle options in Sweave

Description

A small class designed to hold a set of Sweave options

Objects from the Class

Objects can be created by calls of the form `new("SweaveOptions", ...)`.

Slots

options: Object of class "list" A list of strings representing options from a Sweave document.

Methods

show signature(object = "SweaveOptions"): Outputs the options
getOptions signature(object = "SweaveOptions"): Retrieves the options
numOptions signature(object = "SweaveOptions"): Returns the number of options

Author(s)

Jeff Gentry

See Also

[Sweave](#), [codeChunk](#)

Vignette-class *A Class To Represent Vignettes*

Description

This is a class that will represent a vignette file in R, it extends the DynDoc class

Slots

package: Object of class "character" The package that this vignette is associated with
vigPkgVersion: Object of class "VersionNumber" The version number for this vignette's package
indexEntry: Object of class "character", from class "DynDoc" The VignetteIndexEntry field from the document file
title: Object of class "character", from class "DynDoc" The title of the vignette
path: Object of class "character", from class "DynDoc" The path to the vignette file stored locally
pdfPath: Object of class "character", from class "DynDoc" The path to a PDF representation of the vignette
depends: Object of class "character", from class "DynDoc" Any package dependencies for this vignette
requires: Object of class "character", from class "DynDoc" Any requires level dependencies for this vignette
suggests: Object of class "character", from class "DynDoc" Any suggests level dependencies for this vignette
keywords: Object of class "character", from class "DynDoc" Any keywords for this vignette
codeChunks: Object of class "chunkList", from class "DynDoc" A list of code chunks from this vignette

Extends

Class "DynDoc", directly.

Methods

package signature(object = "Vignette"): Retrieves the package name that this vignette is associated with
vigPkgVersion signature(object = "Vignette"): Retrieves the version of the package that this vignette is associated with

Note

The Vignette class is extending the DynDoc class by further associating the DynDoc concepts with a specific R package.

Author(s)

Jeff Gentry

See Also

[DynDoc-class](#), [Sweave](#)

chunkList-class *Class "chunkList"*

Description

This class is essentially a wrapper for the `codeChunk` class. It contains all of the `codeChunks` from a vignette file.

Objects from the Class

Objects can be created by calls of the form `new("chunkList", ...)`.

Slots

chunks: Object of class "list" Stores a list of `codeChunk` objects, representing all of the code chunks from a vignette file.

evalEnv: Object of class "environment" An environment used for evaluation of the code chunks.

Methods

show signature(object = "chunkList"): Displays verbose information about the code chunks

chunks signature(object = "chunkList"): Retrieves a list of `codeChunk` objects

getAllCodeChunks signature(object = "chunkList"): Collapses all of the code chunks into one block of code and returns this

getChunk signature(object = "chunkList"): Retrieves a specific code chunk

numChunks signature(object = "chunkList"): Returns the number of code chunks in this object

setChunk<- signature(object = "chunkList", value="character"): Changes the code in a given `codeChunk` contained by this object

summary signature(object = "chunkList"): A less verbose display of information about the object

evalChunk signature(object = "chunkList", pos="numeric"): Evaluates the code chunk at the specified position in the `chunkList` object

Author(s)

Jeff Gentry

See Also

[Sweave](#), [codeChunk](#), [vignetteCode](#)

Examples

```
library("utils")
testfile <- system.file("Sweave", "Sweave-test-1.Rnw", package="utils")
z <- Stangle(testfile, driver=tangleToR)
```

codeChunk-class *Class "codeChunk"*

Description

A class to wrap necessary information for a code chunk from a vignette file.

Objects from the Class

Objects can be created by calls of the form `new("codeChunk", ...)`.

Slots

chunkName: Object of class "character" The name (if one exists) for the code chunk

chunk: Object of class "character" The code from the code chunk

options: Object of class "SweaveOptions" Any options that were set at the time the code chunk appears in the vignette file

Methods

evalChunk signature(object = "codeChunk", env="environment"): Will evaluate the code in the code chunk using the environment specified. If no environment is specified, `.GlobalEnv` is used.

show signature(object = "codeChunk"): Displays the information for the code chunk

chunk<- signature(object = "codeChunk", value="character"): Edits the chunk slot of the object

chunk signature(object = "codeChunk"): Returns the chunk slot of the object

chunkName signature(object = "codeChunk"): Returns the name of the code chunk

getOptions signature(object = "codeChunk"): Returns the actual options from the options slot.

SweaveOptions signature(object = "codeChunk"): Returns the object stored in the options slot.

Author(s)

Jeff Gentry

See Also

[Sweave](#), [SweaveOptions](#), [chunkList](#)

Examples

```
require("utils")
testfile <- system.file("Sweave", "Sweave-test-1.Rnw", package="utils")
z <- Stangle(testfile, driver=tangleToR)
getChunk(z,1)
```

getPkgVigList *A function to retrieve a listing of package vignettes*

Description

Functionality to retrieve vignette metadata, on a per-vignette or a per-package level.

Usage

```
getPkgVigList(pkg, vigDescFun=baseVigDesc, vigPath = "/doc/",
vigExt="\.(Rnw|Snw|rnw|snw|Rtex)$", pkgVers = TRUE)
```

```
getVigInfo(vig, pkg=NULL, vigDescFun=baseVigDesc, pkgVers=TRUE)
```

Arguments

| | |
|------------|--|
| pkg | Path to a package directory |
| vig | Filename of a vignette |
| vigDescFun | Function to provide output string for display |
| vigPath | Path to directory that contains vignettes in the package |
| vigExt | Regular expression pattern to match vignette file extensions |
| pkgVers | Record the package version with the other vignette metadata |

Details

`getPkgVigList`: This function will look at all vignette files in the directory `<pkg>/<vigPath>`. It will then extract any header information (using `getVigInfo`), and return a list of this information.

`getVigInfo`: This function will retrieve the metadata from a particular vignette file. Any line starting with `'%\Vignette'` is taken to be metadata. Common values include `VignetteIndexEntry` (required), `VignetteKeywords`, `VignetteDepends`, etc. A named list of lists is returned to the user, where the names correspond to the particular metadata variable.

Both functions take a parameter `baseVigDesc`, which is a function to provide the output string to correspond with a vignette summary. This function is directly called by `getVigInfo`. It takes one parameter, which is a `vigInfo` list from `getVigInfo`.

Author(s)

Jeff Gentry

See Also

[vignette](#)

Examples

```
## Not run:
## We need a vignette for this to work
dynPath <- system.file(package="DynDoc")
vigList <- getPkgVigList(dynPath)
vigList

## End (Not run)
```

`getVignette`*A function to handle vignette files*

Description

This function will take a vignette file and return a `Vignette` object in R which can be manipulated further.

Usage

```
getVignette(vigPath, eval = TRUE)
```

Arguments

| | |
|----------------------|--|
| <code>vigPath</code> | The file path of the vignette file |
| <code>eval</code> | Whether or not to evaluate the code chunks |

Details

This function should still be considered experimental

Value

A valid `Vignette` object representing this vignette

Author(s)

Jeff Gentry

See Also

[Vignette-class](#)

getVignetteCode *Functionality to manage code chunks from a vignette*

Description

These functions allow for processing and management of vignette code chunks within R. Users can directly manipulate the code chunks, as well as evaluate them at their option.

Usage

```
getVignetteCode(vigPath, evalEnv = new.env())
editVignetteCode(vigCode, pos, code)
```

Arguments

| | |
|---------|---|
| vigPath | File path of vignette file to process |
| evalEnv | An environment to use for chunk evaluations |
| vigCode | The vignetteCode object to edit |
| pos | The position of the code chunk to edit |
| code | The new code chunk |

Details

getVignetteCode: This function will call `Stangle` using the `tangleToR` driver in order to retrieve the code chunks from the specified vignette file. It will then compile the other pertinent information and return a new `vignetteCode` object.

editVignetteCode: This function will edit a code chunk contained within a `vignetteCode` and return a new object representing that change. The evaluation environment in the new object is a copy of the original as well, *not* the same environment.

Author(s)

Jeff Gentry

See Also

[Sweave](#), [vignetteCode](#), [tangleToR](#)

getVignetteHeader *A function to read vignette header information*

Description

Given a vignette filename, will read in the vignette header metadata.

Usage

```
getVignetteHeader(vig, field)
hasVigHeaderField(vig, field="VignetteIndexEntry")
```


Arguments

| | |
|-------|-----------------------------|
| vig | Vignette filename |
| field | A specific field to extract |

Details

The `getVignetteHeader` function will extract the metadata from a vignette file and return it as a named list, where the names of the list elements correspond to the metadata fields, and the elements themselves the values. If a specific field is desired, it can be specified with the 'field' argument.

The `hasVigHeaderField` function is a simple wrapper around `getVignetteHeader` and will most likely be removed in the very near future. It just is a boolean to report if a given header field exists or not.

Author(s)

Jeff Gentry

tangleToR

An Sweave driver to retrieve code chunks

Description

A driver function for `Sweave` which will provide the user with code chunks from a vignette file within R. Functionality is very similar to that provided by `Stangle` except that an R object is returned as opposed to the chunks being written to a file.

Usage

```
tangleToR()
```

Value

An object of type `chunkList` is returned, which contains the code chunks from the vignette file.

Author(s)

Jeff Gentry

See Also

[Stangle](#), [Sweave](#), [chunkList](#)

Examples

```
require("utils")
testfile <- system.file("Sweave", "Sweave-test-1.Rnw", package="utils")
z <- Stangle(testfile, driver=tangleToR)
```

vignetteCode-class *Class "vignetteCode"*

Description

This class represents the code chunks and other related information from a vignette file. It also provides for the ability to evaluate the code chunks in a separate environment.

Objects from the Class

Objects can be created by calls of the form `new("vignetteCode", ...)`. Also, a helper function `getVignetteCode` is provided that will do all of the dirty work required to retrieve a `vignetteCode` object from a vignette file.

Slots

chunkList: Object of class "chunkList" Holds the code chunks from the vignette file

path: Object of class "character" The path of the vignette file

vigPackage: Object of class "character" The package (if appropriate) that the vignette came from

depends: Object of class "character" Any package dependencies for the vignette

evalEnv: Object of class "environment" An environment used for evaluation of the code chunks.

Methods

show signature(object = "vignetteCode"): Displays information about the code contained in the object

chunkList signature(object = "vignetteCode"): Retrieves the chunkList object.

chunks signature(object = "vignetteCode"): Retrieves the actual code chunks (not wrapped by the chunkList class)

getDepends signature(object = "vignetteCode"): Returns the list of package dependencies for this vignette

evalChunk signature(object = "vignetteCode", pos="numeric"): Will evaluate the specified code chunk in the evalEnv environment

evalEnv signature(object = "vignetteCode"): Returns the evaluation environment

getChunk signature(object = "vignetteCode", pos="numeric"): Returns the codeChunk object representing the specified code chunk position

numChunks signature(object = "vignetteCode"): Returns the number of chunks in the object

vigPackage signature(object = "vignetteCode"): Returns the package the vignette is a part of

path signature(object = "vignetteCode"): Returns the local file path to the vignette

setChunk<- signature(object = "vignetteCode", pos="numeric", value="character"): Resets the code chunk specified by pos to contain the code specified by value

summary signature(object = "vignetteCode"): A less verbose output of information than with show

Author(s)

Jeff Gentry

See Also

[Sweave](#), [getVignetteCode](#), [editVignetteCode](#), [chunkList](#)

Index

*Topic classes

chunkList-class, 4
codeChunk-class, 5
DynDoc-class, 1
SweaveOptions-class, 2
Vignette-class, 3
vignetteCode-class, 10

*Topic utilities

getPkgVigList, 6
getVignette, 7
getVignetteCode, 8
getVignetteHeader, 8
tangleToR, 9

baseVigDesc (*getPkgVigList*), 6

chunk (*codeChunk-class*), 5
chunk, *codeChunk-method*
 (*codeChunk-class*), 5
chunk<- (*codeChunk-class*), 5
chunk<-, *codeChunk-method*
 (*codeChunk-class*), 5
chunkList, 5, 9, 11
chunkList (*chunkList-class*), 4
chunkList, *vignetteCode-method*
 (*vignetteCode-class*), 10
chunkList-class, 4
chunkName (*codeChunk-class*), 5
chunkName, *codeChunk-method*
 (*codeChunk-class*), 5
chunks (*chunkList-class*), 4
chunks, *chunkList-method*
 (*chunkList-class*), 4
chunks, *DynDoc-method*
 (*DynDoc-class*), 1
chunks, *vignetteCode-method*
 (*vignetteCode-class*), 10
codeChunk, 2, 4
codeChunk (*codeChunk-class*), 5
codeChunk, *SweaveOptions-method*
 (*SweaveOptions-class*), 2
codeChunk-class, 5
codeChunks (*DynDoc-class*), 1

codeChunks, *DynDoc-method*
 (*DynDoc-class*), 1

DynDoc-class, 4
DynDoc-class, 1

editVignetteCode, 11
editVignetteCode
 (*getVignetteCode*), 8
evalChunk (*vignetteCode-class*), 10
evalChunk, *chunkList-method*
 (*chunkList-class*), 4
evalChunk, *codeChunk-method*
 (*codeChunk-class*), 5
evalChunk, *DynDoc-method*
 (*DynDoc-class*), 1
evalChunk, *vignetteCode-method*
 (*vignetteCode-class*), 10
evalEnv (*vignetteCode-class*), 10
evalEnv, *chunkList-method*
 (*chunkList-class*), 4
evalEnv, *vignetteCode-method*
 (*vignetteCode-class*), 10

getAllCodeChunks
 (*chunkList-class*), 4
getAllCodeChunks, *chunkList-method*
 (*chunkList-class*), 4
getChunk (*chunkList-class*), 4
getChunk, *chunkList-method*
 (*chunkList-class*), 4
getChunk, *DynDoc-method*
 (*DynDoc-class*), 1
getChunk, *vignetteCode-method*
 (*vignetteCode-class*), 10
getDepends (*vignetteCode-class*),
 10
getDepends, *DynDoc-method*
 (*DynDoc-class*), 1
getDepends, *vignetteCode-method*
 (*vignetteCode-class*), 10
getKeywords (*DynDoc-class*), 1
getKeywords, *DynDoc-method*
 (*DynDoc-class*), 1

- getOptions (*SweaveOptions*-class),
2
- getOptions, codeChunk-method
(*codeChunk*-class), 5
- getOptions, SweaveOptions-method
(*SweaveOptions*-class), 2
- getPkgVigList, 6
- getRequires (*DynDoc*-class), 1
- getRequires, DynDoc-method
(*DynDoc*-class), 1
- getSuggests (*DynDoc*-class), 1
- getSuggests, DynDoc-method
(*DynDoc*-class), 1
- getVigInfo (*getPkgVigList*), 6
- getVigInfoNames (*getPkgVigList*), 6
- getVignette, 7
- getVignetteCode, 8, 11
- getVignetteHeader, 8

- hasVigHeaderField
(*getVignetteHeader*), 8

- indexEntry (*DynDoc*-class), 1
- indexEntry, DynDoc-method
(*DynDoc*-class), 1

- numChunks (*chunkList*-class), 4
- numChunks, chunkList-method
(*chunkList*-class), 4
- numChunks, DynDoc-method
(*DynDoc*-class), 1
- numChunks, vignetteCode-method
(*vignetteCode*-class), 10
- numOptions (*SweaveOptions*-class),
2
- numOptions, SweaveOptions-method
(*SweaveOptions*-class), 2

- path (*vignetteCode*-class), 10
- path, DynDoc-method
(*DynDoc*-class), 1
- path, vignetteCode-method
(*vignetteCode*-class), 10
- pdfPath (*DynDoc*-class), 1
- pdfPath, DynDoc-method
(*DynDoc*-class), 1
- print.pkgFileList
(*getPkgVigList*), 6

- setChunk (*chunkList*-class), 4
- setChunk<- (*chunkList*-class), 4
- setChunk<-, chunkList-method
(*chunkList*-class), 4
- setChunk<-, DynDoc-method
(*DynDoc*-class), 1
- setChunk<-, vignetteCode-method
(*vignetteCode*-class), 10
- show, chunkList-method
(*chunkList*-class), 4
- show, codeChunk-method
(*codeChunk*-class), 5
- show, DynDoc-method
(*DynDoc*-class), 1
- show, SweaveOptions-method
(*SweaveOptions*-class), 2
- show, vignetteCode-method
(*vignetteCode*-class), 10
- Stangle, 9
- summary, chunkList-method
(*chunkList*-class), 4
- summary, DynDoc-method
(*DynDoc*-class), 1
- summary, vignetteCode-method
(*vignetteCode*-class), 10
- Sweave, 2, 4, 5, 8, 9, 11
- SweaveOptions, 5
- SweaveOptions
(*SweaveOptions*-class), 2
- SweaveOptions, codeChunk-method
(*codeChunk*-class), 5
- SweaveOptions-class, 2
- tangleToR, 8, 9
- tangleToRFinish (*tangleToR*), 9
- tangleToRRuncode (*tangleToR*), 9
- tangleToRSetup (*tangleToR*), 9
- transformVigInfoLine
(*getPkgVigList*), 6

- vignette, 6
- Vignette-class, 7
- Vignette-class, 3
- vignetteCode, 4, 8
- vignetteCode
(*vignetteCode*-class), 10
- vignetteCode-class, 10
- vigPackage (*vignetteCode*-class),
10
- vigPackage, Vignette-method
(*Vignette*-class), 3
- vigPackage, vignetteCode-method
(*vignetteCode*-class), 10
- vigPkgVersion (*Vignette*-class), 3
- vigPkgVersion, Vignette-method
(*Vignette*-class), 3