Package 'review'

July 2, 2014

Type Package

Title Manage Review Logs

Version 2.5

Date 2013-04-18

Author Tim Bergsma

Maintainer Tim Bergsma <timb@metrumrg.com>

Depends XML, Hmisc

Suggests

Description Functions for managing logs of reviews of subversioned files (http://subversion.apache.org/).

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-18 22:49:09

R topics documented:

`review-package` *Manage Review Logs*

Description

Functions for managing logs of reviews of subversioned files (http://subversion.apache.org/).

Details

Package:	review
Type:	Package
Version:	2.5
Date:	2013-04-18
License:	GPL-2
LazyLoad:	yes

These functions are rather flexible, but convenient defaults are chosen. Easiest usage is described here.

Making a Log

1. In R, navigate to the check-out directory with `setwd()`.
2. Type `logCreate()`.
3. Be sure to add your log file and check it in with svn tools.

Making Assignments

1. In R, navigate to the check-out directory with `setwd()`.
2. Try `dir(recursive=TRUE)` to get a list of possible assignments. customize as necessary.
3. To assign all files to "anyone", type `logAssign(dir(recursive=TRUE))`.

4. To assign just the files in a directory, navigate there and type `logAssign()`.
5. To review the result, type `logRead()`.
6. To undo the result, type `logRevert()`.
7. Check in your changes using svn tools.

Reviewing Files

1. Update your checkout with svn tools.
2. Naviate to the directory containing the log, or below.
3. View your assignments with `logAssignments()` or `logAssignments(reviewer = 'anyone')`.
4. Capture the name of a particular file, e.g. `f <- logAssignments()[[1]]`.
5. To approve the file, say `logAccept(f)`.
6. Check in your changes using svn tools.

Summarizing

1. Navigate to the directory containing the log, or below.
2. Type `logSummary()`.
3. If `logPending()` is empty, all assigned reviews are complete.

Shortcut

1. Technically, it is not necessary that a file ever be assigned. The first acceptance of an existing file suffices (origin must be specified, however). The only two essential functions are `logCreate()` and `logAccept()`.

Author(s)

Tim Bergsma

Maintainer: Tim Bergsma <timb@metrumrg.com>

References

Development supported by Metrum Institute: http://metruminstitute.org.

See Also

- absDir
- parentDir
- logName
- logCreate
- logRoot
- repoInfo
- revision
- author

- svnDate
- gmt
- logRead
- logWrite
- relPath
- logTarget
- logQueue
- logAppend
- logAssign
- logAccept
- logSummary
- logPending
- logAssignments
- logAppendix

---

absDir *Absolute Directory*

Description

Return the full path of a directory.

Usage

```
absDir(directory)
```

Arguments

directory	an absolute or relative path of a directory.

Details

Relative paths include subdirectories, "..",".", "~","~/", etc. absDir() should return the complete path, back to the volume root.

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- relPath

`author` *Subversion Author of a File*

Description

Gives the author of a file that is under subversion control.

Usage

```
author(file = logRoot())
```

Arguments

`file`	a (path to a) subversioned file

Details

If the file is not subversioned, NA is returned. This function is not vectorized, so use sapply for multiple files.

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- revision
- repoInfo

gmt *Current GMT Time*

Description

Display current UTC (GMT).

Usage

```
gmt()
```

Details

Sys.time() returns POSIXlt, which loses time zone when coerced to character. This function makes the necessary repair.

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logQueue

logAccept *Approve a File*

Description

Indicate acceptance of a file by logging an entry with nonzero version.

Usage

```
logAccept(
file=dir(),
directory=getwd(),
origin=logOrigin(file,directory),
reviewer=Sys.info()['user'],
force=FALSE,
...
)
```

Arguments

file	one or more files to accept
directory	the parent of file
origin	the file from which 'file' originates
reviewer	the user issuing the opinion, normally oneself
force	if false (default) nonexistent files cause an error
...	arguments passed to logQueue

Details

Can be undone with logRevert(). Calls logAppend(), which gives an error if origin is NA (default) and no precedent exists.

Value

(invisible) number of records created.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logAssign
- logQueue
- logRoot

logAppend *Append a Review Log*

Description

Add records to a review log.

Usage

```
logAppend(new,directory = logRoot(),  ...)
logOrigin(file,directory = logRoot(), ...)
```

Arguments

new	a data.frame of new records to append, as created with logQueue().
directory	the directory containing the log, or any directory below it
file	vector of file names.
...	arguments passed to logQueue().

Details

Generally these functions are not called by the user. Check "see also" for preferred interface.

Value

logAppend: (invisible) the number of rows added. logOrigin: the precedents for file origin, if any, or file itself where precedent not available.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logAssign
- logQueue
- logAccept

logAppendix *Convert a Log Summary to a Report-ready Table*

Description

Converts the output of logSummary to a Latex table, suitable for inclusion in a report, e.g. as an appendix. Gives a warning if logPending() has rows.

Usage

```
logAppendix(
object=logSummary()[,c('file','revf','reviewer','time')],
dir=getwd(),
file=file.path(dir,'review.tex'),
label='review',
rowname=NULL,
caption='Summary of file review: file, last version reviewed (revf), system identifier of reviewer, ar
caption.lot='Summary of file review',
where='H',
show=TRUE,
...
)
```

Arguments

object	object to be plotted
dir	a directory used to build the default file path
file	a file path for the output
label	a character string used for the tex label
rowname	default NULL suppresses rownames
caption	table caption
caption.lot	caption variant for use in list of tables
where	default H ("here") places table exactly where specified
show	default TRUE invokes the dvi viewer
...	passed to latex

Details

This function calls latex in package Hmisc. All arguments except dir are passed to latex. dir is just a convenience argument to place the output in other than the current working directory (using the default file name). It is ignored if file is supplied.

Value

used for side effects

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logSummary

`logAssign` *Assign Review Tasks*

Description

Appends a log with defaults supplied by logQueue(). These are appropriate for assigments. Default reviewer is set to "anyone". Default revision is zero, which is diagnostic of an assignment.

Usage

```
logAssign(file = dir(), directory = getwd(), origin = file, ...)
```

Arguments

`file`	a character vector of filenames (paths) relative to directory
`directory`	the parent of 'file'
`origin`	a character vector of filenames (paths) relative to directory, from which 'file' originates
`...`	arguments passed to logQueue()

Details

Since files with revision of zero normally do not exist, no acceptance is implied by an assigned record. This function may be used to specify a reviewer for certain files. Default reviewer "anyone" simply declares that an entity should be reviewed. "Origin" files may also be specified, indicating that a file originates from (e.g. is created by) some other file. The function logPending() identifies files needing (further) review. A file's origin is itself, by default. Currently, logAssign() calls logAppend(), which allows NA for origin (defaults to precedent or to file itself).

Value

(invisible) the number of records created.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logAssignments
- logAppend
- logQueue
- logAccept

`logAssignments` *Detect Files to be Reviewed*

Description

Gives a vector of the pending files assigned to reviewer (normally oneself).

Usage

```
logAssignments(directory = getwd(), reviewer = Sys.info()[["user"]])
```

Arguments

`directory`	the directory containing the log, or any directory below it
`reviewer`	a user name on which to filter results

Details

"Pending" has the meaning described for logPending(). "Assigned" has the meaning described for logAssign(). A character vector is returned, from which elements can be selected for use as arguments to logAccept().

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logAssign
- logPending
- logAccept

---

logCreate *Create a Review Log*

Description

Creates an empty table in the directory specified with name "QClog.csv".

Usage

```
logCreate(directory = getwd())
```

Arguments

directory	directory in which to create the log file

Details

Columns are as follows.

file the path and name of the accepted file, relative to the specified directory, and not beginning with "/"

origin identical to `file` or the name of some other file responsible for this content

revf the reviewed revision of `file`

revo the review-time version of `file`'s origin

reviewer the R user name of the reviewer of `file`

time date and time (GMT) of acceptance of `file`

Value

used for side effects.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logName
- logAssign
- logAccept

`logName` *The Full Path and Name of the Log File*

Description

Gives a full-path log name for the directory supplied. The default directory is logRoot(), in which case the file must already exist.

Usage

```
logName(directory = logRoot())
```

Arguments

`directory`	a directory for which a log name is desired

Details

Without arguments, returns full path to log file when current directory is at or below the root (directory containing the log).

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logRoot
- absDir
- logCreate

`logPending` *List Log Entries that Require Review*

Description

Checks the log summary and returns entries `headf > revf` or `heado > revo`. That is, an item is pending if there exists a newer revision of its file or origin.

Usage

```
logPending(directory = getwd())
```

Arguments

`directory`	any directory at or below that containing the log.

Details

One always wants to review the latest revision of a file, given the latest revision of its origin. Typically a file is its own origin (e.g. a directly-authored script), in which case it suffices to verify that there is no newer revision. For derived objects, the latest revision of the origin is informative, and is independent of the file revision. Further review is indicated if either the file, its origin, or both have been revised. If a file has changed but not its origin, an explanation is needed. If a file is the same after a change to its origin, confirmation is needed. It is not unusual for both a file and its origin to change.

Value

data.frame

Note

Be sure to update your checkout for the latest pending information.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- `logSummary`
- `logAssignments`
- `logAssign`
- `logAccept`

logQueue *Prepare Records for Review Log*

Description

Create one or more records with the proper format for appending to a review log.

Usage

```
logQueue(
file=dir(),
directory=getwd(),
origin=file,
revf=0,
revo=0,
reviewer="anyone",
time=gmt(),
force=FALSE
)
```

Arguments

file	one or more files for which to create records
directory	the parent of file
origin	file(s) from which 'file' originates
revf	the svn revision of the file being reviewed
revo	the svn revision of the origin of file
reviewer	the user issuing the opinion, normally oneself
time	character string: date and time of acceptance, using GMT (UTC)
force	if false (the default) nonexistent files cause an error

Details

The required format for a review log is described in logCreate. Functions like logAssign() and logAccept() send output to the log. The function logQueue() creates a data.frame that can be viewed, assigned, and manipulated prior to appending the log. You may want to use logQueue() in conjunction with logAppend(). That is exactly what logAssign() and logAccept() do.

Note that time should be character, each element ending with "GMT".

Value

data.frame, with a format much like QClog.csv

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logCreate
- logAssign
- logAccept
- logAppend

---

logRead *Read a Review Log*

Description

Read a review log from storage and create a corresponding data.frame.

Usage

```
logRead(directory = getwd())
```

Arguments

directory	the directory containing the log, or any directory below it

Details

Regardless of the current working directory, if the directory containing the log is an ancestor (parent, grandparent, etc.) the log will be found and read. It is an error if the directory does not exist or if the log does not exist, or if the log is styled for package version 1.5 and earlier.

Value

data.frame with format much like QClog.csv

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logCreate
- logRoot
- logAppend
- logWrite

`logRoot` *Identify the Directory Containing the Review Log*

Description

This function searches backward from the current working directory until it finds the directory containing the review log. If no log is found, NULL is returned. It is an error if the directory argument does not exist.

Usage

```
logRoot(directory = getwd())
```

Arguments

`directory`	the directory containing the log, or any directory below it

Details

This is a recursive function.

Value

character: the full path and name of the directory containing the review log.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- parentDir
- logName
- logCreate
- logAppend

`logSummary` *Summarize a Review Log*

Description

Summarize the review log associated with a particular directory.

Usage

```
logSummary(directory = getwd())
```

Arguments

`directory`	the directory containing the log, or any directory below it

Details

Whereas the review log can have any number of records for the same file, the summary reduces the data to one record per file. It is an error if any element of the file column is missing.

The log is read, then sorted on file, revision, time. The last record for each file is retained.

The function logSummary() uses the svn interface to retrieve the latest ('head') revision of each file and origin, appending them as `headf` and `heado` respectively. Column order is controlled.

print.logSummary() tries to eliminate some redundant information for easier viewing. Columns `origin`, `revo`, and `heado` are marked as " where non-informative, and are dropped if entirely non-informative.

Value

data.frame

Note

Be sure to update your checkout for the most current summary.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- logPending
- logAssignments
- logRead
- logTarget
- svnDate

---

logTarget *Generate Fully Qualified Filenames*

Description

Prepend the absolute file path to each file name, and test for existence.

Usage

```
logTarget(file = dir(),directory = getwd(), force = FALSE)
```

Arguments

file	one or more files for which to create targets
directory	the parent of file
force	if false (the default) nonexistent files cause an error

Details

This function is vectorized.

Value

character vector of file names

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- absDir
- logQueue

`logWrite` *Write a Review Log to Storage in Standard Format*

Description

Write a review log, using the same formatting conventions as logRead().

Usage

```
logWrite(x, file)
```

Arguments

`x`	a data.frame, like the output of logQueue()
`file`	a file name

Details

Calls write.table() with quotes and rownames turned off. Separator is comma, NA's are dots.

Normally the user should not have to call logWrite(). The functions logCreate(), logAppend(), and logRevert() call it.

Value

Used for side effects.

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- `logRead`
- `logCreate`
- `logAppend`
- `help`

parentDir *Identify the Parent of a Directory*

Description

Generate a fully qualified directory name for the parent of the specified directory.

Usage

```
parentDir(directory = getwd())
```

Arguments

directory	the directory whose parent is sought

Details

It is an error if the directory does not exist. If it exists but has no parent, NULL is returned. The path includes the volume root. This function is not vectorized.

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- absDir
- relPath
- logRoot

`relPath` *Create File Paths Relative to the Log Root*

Description

For the files specified by the arguments, create file paths relative to the directory containing the review log.

Usage

```
relPath(file = dir(), directory = getwd())
```

Arguments

`file`	one or more files for which to create paths
`directory`	the parent directory of file

Details

The elements of 'file' must be encoded relative to 'directory'.

Value

character vector

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- `parentDir`
- `absDir`
- `logRoot`
- `help`

repoInfo *Retrieve Repository Information*

Description

For a given file, retrieve svn repository information as XML.

Usage

```
repoInfo(file = logRoot())
```

Arguments

file	a file name

Details

Not normally called by the user.

Value

an XML representation of `svn info`

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- revision
- author
- logRoot

`revision` *Subversion Revision of a File*

Description

Gives the revision of a file that is under subversion control.

Usage

```
revision(file = logRoot())
```

Arguments

`file`	a (path to a) subversioned file

Details

If the file is not subversioned, NA is returned. This function is not vectorized, so use sapply for multiple files. The file name will be single-quoted if it contains a space and does not start with single or double quote.

Value

numeric

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- `author`
- `repoInfo`

svnDate *Subversion Date of a File*

Description

Gives the date and time of a file that is under subversion control, with GMT as the timezone (UTC).

Usage

```
svnDate(file = logRoot())
```

Arguments

file	a (path to a) subversioned file

Details

If the file is not subversioned, NA is returned. This function is not vectorized, so use sapply for multiple files.

Value

character

Author(s)

Tim Bergsma

References

http://metruminstitute.org

See Also

- revision
- repoInfo

Index