

---+ Package TWiki::Store::RcsFile

This class is PACKAGE PRIVATE to Store, and should never be used from anywhere else. It is the base class of implementations of stores that manipulate RCS format files.

The general contract of the methods on this class and its subclasses calls for errors to be signalled by Error::Simple exceptions.

Refer to Store.pm for models of usage.

## **ClassMethod new (\$session, \$web, \$topic, \$attachment)**

Constructor. There is one object per stored file.

Note that \$web, \$topic and \$attachment must be untainted!

## **ObjectMethod finish ()**

Break circular references.

## **ObjectMethod getRevisionInfo (\$version) -> (\$rev, \$date, \$user, \$comment)**

- \$version if 0 or undef, or out of range (version number > number of revs) will return info about the latest revision.

Returns (rev, date, user, comment) where rev is the number of the rev for which the info was recovered, date is the date of that rev (epoch s), user is the login name of the user who saved that rev, and comment is the comment associated with the rev.

Designed to be overridden by subclasses, which can call up to this method if file-based rev info is required.

## **ObjectMethod getLatestRevision () -> \$text**

Get the text of the most recent revision

## **ObjectMethod getLatestRevisionTime () -> \$text**

Get the time of the most recent revision

## **ObjectMethod getWorkArea (\$key) -> \$directorypath**

Gets a private directory uniquely identified by \$key. The directory is intended as a work area for plugins.

The standard is a directory named the same as "key" under \$TWiki::cfg{WorkingDir}/work\_areas

## **ObjectMethod getTopicNames () -> @topics**

Get list of all topics in a web

- `$web` - Web name, required, e.g. 'Sandbox'

Return a topic list, e.g. ( 'WebChanges', 'WebHome', 'WebIndex', 'WebNotify' )

## ObjectMethod `getWebNames` () -> @webs [↗](#)

Gets a list of names of subwebs in the current web

## ObjectMethod `searchInWebContent`

(`$searchString`, `$web`, `\@topics`, `\%options`) -> `\%map`

Search for a string in the content of a web. The search must be over all content and all formatted meta-data, though the latter search type is deprecated (use `searchMetaData` instead).

- `$searchString` - the search string, in egrep format if regex
- `$web` - The web to search in
- `\@topics` - reference to a list of topics to search
- `\%options` - reference to an options hash

The `\%options` hash may contain the following options:

- `type` - if `regex` will perform a egrep-syntax RE search (default "")
- `casesensitive` - false to ignore case (default true)
- `files_without_match` - true to return files only (default false)

The return value is a reference to a hash which maps each matching topic name to a list of the lines in that topic that matched the search, as would be returned by 'grep'. If `files_without_match` is specified, it will return on the first match in each topic (i.e. it will return only one match per topic, and will not return matching lines).

## ObjectMethod `searchInWebMetaData` (`$query`, `\@topics`) -> `\%matches`

Search for a meta-data expression in the content of a web. `$query` must be a `Twiki::Query` object.

Returns a reference to a hash that maps the names of topics that all matched to the result of the query expression (e.g. if the query expression is 'TOPICPARENT.name' then you will get back a hash that maps topic names to their parent.

SMELL: this is **really** inefficient!

## ObjectMethod `moveWeb` (`$newWeb`)

Move a web.

## ObjectMethod `getRevision` (`$version`) -> `$text`

Get the text for a given revision. The version number must be an integer.

**Virtual method** - must be implemented by subclasses

## **ObjectMethod storedDataExists () -> \$boolean**

Establishes if there is stored data associated with this handler.

## **ObjectMethod getTimestamp () -> \$integer**

Get the timestamp of the file Returns 0 if no file, otherwise epoch seconds

## **ObjectMethod restoreLatestRevision (\$user)**

Restore the plaintext file from the revision at the head.

## **ObjectMethod removeWeb (\$web)**

- \$web - web being removed

Destroy a web, utterly. Removed the data and attachments in the web.

Use with great care! No backup is taken!

## **ObjectMethod moveTopic (\$newWeb, \$newTopic)**

Move/rename a topic.

## **ObjectMethod copyTopic (\$newWeb, \$newTopic)**

Copy a topic.

## **ObjectMethod moveAttachment**

**(\$newWeb, \$newTopic, \$newAttachment)**

Move an attachment from one topic to another. The name is retained.

## **ObjectMethod copyAttachment (\$newWeb, \$newTopic)**

Copy an attachment from one topic to another. The name is retained.

## **ObjectMethod isAsciiDefault () -> \$boolean**

Check if this file type is known to be an ascii type file.

## **ObjectMethod setLock (\$lock, \$user)**

Set a lock on the topic, if \$lock, otherwise clear it. \$user is a wikiname.

SMELL: there is a tremendous amount of potential for race conditions using this locking approach.

## **ObjectMethod isLocked () -> (\$user, \$time)**

See if a twiki lock exists. Return the lock user and lock time if it does.

## **ObjectMethod setLease (\$lease)**

- `$lease` reference to lease hash, or undef if the existing lease is to be cleared.

Set an lease on the topic.

## **ObjectMethod getLease () -> \$lease**

Get the current lease on the topic.

## **ObjectMethod removeSpuriousLeases (\$web)**

Remove leases that are not related to a topic. These can get left behind in some store implementations when a topic is created, but never saved.

## **ObjectMethod getStream () -> \\*STREAM**

Return a text stream that will supply the text stored in the topic.

## **ObjectMethod numRevisions () -> \$integer**

Must be provided by subclasses.

Find out how many revisions there are. If there is a problem, such as a nonexistent file, returns 0.

**Virtual method** - must be implemented by subclasses

## **ObjectMethod initBinary ()**

Initialise a binary file.

Must be provided by subclasses.

**Virtual method** - must be implemented by subclasses

## **ObjectMethod initText ()**

Initialise a text file.

Must be provided by subclasses.

**Virtual method** - must be implemented by subclasses

## ObjectMethod addRevisionFromText

(**\$text**, **\$comment**, **\$user**, **\$date**)

Add new revision. Replace file with text.

- `$text` of new revision
- `$comment` checkin comment
- `$user` is a wikiname.
- `$date` in epoch seconds; may be ignored

**Virtual method** - must be implemented by subclasses

## ObjectMethod addRevisionFromStream

(**\$fh**, **\$comment**, **\$user**, **\$date**)

Add new revision. Replace file with contents of stream.

- `$fh` filehandle for contents of new revision
- `$comment` checkin comment
- `$user` is a wikiname.
- `$date` in epoch seconds; may be ignored

**Virtual method** - must be implemented by subclasses

## ObjectMethod replaceRevision

(**\$text**, **\$comment**, **\$user**, **\$date**)

Replace the top revision.

- `$text` is the new revision
- `$date` is in epoch seconds.
- `$user` is a wikiname.
- `$comment` is a string

**Virtual method** - must be implemented by subclasses

## ObjectMethod deleteRevision ()

Delete the last revision - do nothing if there is only one revision

**Virtual method** - must be implemented by subclasses

## ObjectMethod revisionDiff (**\$rev1**, **\$rev2**, **\$contextLines**) -> **\@diffArray**

`rev2` newer than `rev1`. Return reference to an array of [ `diffType`, `$right`, `$left` ]

**Virtual method** - must be implemented by subclasses

!!!getRevision!!!

ObjectMethod addRevisionFromText (`$text`,`$comment`,`$user`,`$date`)

## ObjectMethod **getRevisionAtTime** (**\$time**) -> **\$rev**

Get a single-digit version number for the rev that was alive at the given epoch-secs time, or undef if none could be found.

**Virtual method** - must be implemented by subclasses

## ObjectMethod **getAttachmentAttributes**

(**\$web**, **\$topic**, **\$attachment**)

returns [stat] for any given web, topic, \$attachment SMELL - should this return a hash of arbitrary attributes so that SMELL + attributes supported by the underlying filesystem are supported SMELL + (eg: windows directories supporting photo "author", "dimension" fields)

## ObjectMethod **getAttachmentList** (**\$web**, **\$topic**)

returns { } of filename => { key => value, key2 => value } for any given web, topic Ignores files starting with \_ or ending with ,v

## ObjectMethod **stringify** ()

Generate string representation for debugging

## ObjectMethod **recordChange** (**\$user**, **\$rev**, **\$more**)

Record that the file changed

## ObjectMethod **eachChange** (**\$since**) -> **\$iterator**

Return iterator over changes - see Store for details

---

This topic: TWiki21Nov > TWikiStoreRcsFileDotPm

Topic revision: r4 - 2008-01-22 - TWikiContributor



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback