NAME

Mail::Box::Mbox − handle folders in Mbox format

INHERITANCE

```
Mail::Box::Mbox
  is a Mail::Box::File
  is a Mail::Box
  is a Mail::Reporter
```

SYNOPSIS

```
use Mail::Box::Mbox;
my $folder = Mail::Box::Mbox->new(folder => $ENV{MAIL}, ...);
```

DESCRIPTION

This documentation describes how Mbox mailboxes work, and also describes what you can do with the Mbox folder object Mail::Box::Mbox.

Extends ‘‘DESCRIPTION’’ in Mail::Box::File.

OVERLOADED

Extends ‘‘OVERLOADED’’ in Mail::Box::File.

overload: **“”**
Inherited, see ‘‘OVERLOADED’’ in Mail::Box

overload: **@{}**
Inherited, see ‘‘OVERLOADED’’ in Mail::Box

overload: **cmp**
Inherited, see ‘‘OVERLOADED’’ in Mail::Box

METHODS

Extends ‘‘METHODS’’ in Mail::Box::File.

Constructors

Extends ‘‘Constructors’’ in Mail::Box::File.

Mail::Box::Mbox->**new**(%options)

-Option	--Defined in	--Default
access	Mail::Box	'r'
body_delayed_type	Mail::Box	Mail::Message::Body::Delayed
body_type	Mail::Box::File	<see description>
coerce_options	Mail::Box	[]
create	Mail::Box	<false>
extract	Mail::Box	10240
field_type	Mail::Box	undef
fix_headers	Mail::Box	<false>
folder	Mail::Box	$ENV{MAIL}
folderdir	Mail::Box	$ENV{HOME}.'/Mail'
head_delayed_type	Mail::Box	Mail::Message::Head::Delayed
head_type	Mail::Box	Mail::Message::Head::Complete
keep_dups	Mail::Box	<false>
lock_extension	Mail::Box::File	'.lock'
lock_file	Mail::Box	<foldername><lock-extension>
lock_timeout	Mail::Box	1 hour
lock_type	Mail::Box	Mail::Box::Locker::DotLock
lock_wait	Mail::Box	10 seconds
locker	Mail::Box	undef
log	Mail::Reporter	'WARNINGS'
manager	Mail::Box	undef
message_type	Mail::Box	Mail::Box::Mbox::Message

```
multipart_type       Mail::Box        Mail::Message::Body::Multipart
remove_when_empty    Mail::Box        <true>
save_on_exit         Mail::Box        <true>
subfolder_extension                   '.d'
trace                Mail::Reporter   'WARNINGS'
trusted              Mail::Box        <depends on folder location>
write_policy         Mail::Box::File  undef
```

access => MODE
body_delayed_type => CLASS
body_type => CLASS|CODE
coerce_options => ARRAY
create => BOOLEAN
extract => INTEGER | CODE | METHOD | 'LAZY'|'ALWAYS'
field_type => CLASS
fix_headers => BOOLEAN
folder => FOLDERNAME
folderdir => DIRECTORY
head_delayed_type => CLASS
head_type => CLASS
keep_dups => BOOLEAN
lock_extension => FILENAME|STRING
lock_file => FILENAME
lock_timeout => SECONDS
lock_type => CLASS|STRING|ARRAY
lock_wait => SECONDS
locker => OBJECT
log => LEVEL
manager => MANAGER
message_type => CLASS
multipart_type => CLASS
remove_when_empty => BOOLEAN
save_on_exit => BOOLEAN
subfolder_extension => STRING

Mbox folders do not support sub-folders. However, this module can simulate sub-directories if the user wants it to. When a subfolder of folder `xyz` is created, we create a directory which is called `xyz.d` to contain them. This extension `.d` can be changed using this option.

trace => LEVEL
trusted => BOOLEAN
write_policy => 'REPLACE'|'INPLACE'|undef

The folder

Extends "The folder" in Mail::Box::File.

`$obj`->**addMessage**($message, `%options`)
Inherited, see "The folder" in Mail::Box

`$obj`->**addMessages**(@messages)
Inherited, see "The folder" in Mail::Box

Mail::Box::Mbox->**appendMessages**(%options)
Inherited, see "METHODS" in Mail::Box::File

`$obj`->**close**(%options)
Inherited, see "The folder" in Mail::Box

`$obj->`**copyTo**($folder, `%options`)
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**delete**(%options)
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**filename**()
Inherited, see ‘‘The folder’’ in Mail::Box::File

`$obj->`**folderdir**([$directory])
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**name**()
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**organization**()
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**size**()
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**type**()
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**update**(%options)
Inherited, see ‘‘The folder’’ in Mail::Box

`$obj->`**url**()
Inherited, see ‘‘The folder’’ in Mail::Box

Folder flags
Extends ‘‘Folder flags’’ in Mail::Box::File.

`$obj->`**access**()
Inherited, see ‘‘Folder flags’’ in Mail::Box

`$obj->`**isModified**()
Inherited, see ‘‘Folder flags’’ in Mail::Box

`$obj->`**modified**([BOOLEAN])
Inherited, see ‘‘Folder flags’’ in Mail::Box

`$obj->`**writable**()
Inherited, see ‘‘Folder flags’’ in Mail::Box

The messages
Extends ‘‘The messages’’ in Mail::Box::File.

`$obj->`**current**([$number|$message|$message_id])
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**find**($message_id)
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**findFirstLabeled**(`$label`, [BOOLEAN, [$msgs]])
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**message**(`$index`, [$message])
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**messageId**(`$message_id`, [$message])
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**messageIds**()
Inherited, see ‘‘The messages’’ in Mail::Box

`$obj->`**messages**(<'ALL'|$range|'ACTIVE'|'DELETED'|$label| !$label|$filter>)
Inherited, see "The messages" in Mail::Box

`$obj->`**nrMessages**(%options)
Inherited, see "The messages" in Mail::Box

`$obj->`**scanForMessages**($message, `$message_ids`, `$timespan`, `$window`)
Inherited, see "The messages" in Mail::Box

Sub-folders

Extends "Sub-folders" in Mail::Box::File.

`$obj->`**listSubFolders**(%options)
Mail::Box::Mbox->**listSubFolders**(%options)

-Option	--Defined in	--Default
check	Mail::Box	<false>
folder	Mail::Box	<from calling object>
folderdir	Mail::Box	<from folder>
skip_empty	Mail::Box	<false>
subfolder_extension		<from object>

check => BOOLEAN
folder => FOLDERNAME
folderdir => DIRECTORY
skip_empty => BOOL
subfolder_extension => STRING
When the method is called on an open folder, the extension defined by it is used to detect sub-folders by default. Otherwise, `'.d'` is tak en.

`$obj->`**nameOfSubFolder**(`$subname`, [$parentname])
Mail::Box::Mbox->**nameOfSubFolder**(`$subname`, [$parentname])
Inherited, see "Sub-folders" in Mail::Box

`$obj->`**openRelatedFolder**(%options)
Inherited, see "Sub-folders" in Mail::Box

`$obj->`**openSubFolder**($subname, `%options`)
Inherited, see "Sub-folders" in Mail::Box

`$obj->`**topFolderWithMessages**()
Mail::Box::Mbox->**topFolderWithMessages**()
Inherited, see "Sub-folders" in Mail::Box

Internals

Extends "Internals" in Mail::Box::File.

`$obj->`**coerce**($message, `%options`)
Inherited, see "Internals" in Mail::Box

`$obj->`**create**($foldername, `%options`)
Mail::Box::Mbox->**create**($foldername, `%options`)

-Option	--Defined in	--Default
folderdir	Mail::Box	undef
subfolder_extension		undef

folderdir => DIRECTORY
subfolder_extension => STRING
If a directory is found on the location of the folder to be created, this STRING is used to extend that directory name with. This will cause the directory to be seen as sub-folder for the created folder. This argument is passed to **folderToFilename()**.

`$obj->`**determineBodyType**($message, `$head`)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**folderToFilename**(`$foldername`, `$folderdir`, [$extension])
Mail::Box::Mbox->**folderToFilename**(`$foldername`, `$folderdir`, [$extension])
Translate a folder name into a filename, using the `$folderdir` value to replace a leading `=`. If no `$extension` is specified and this method is called as instance method, new(subfolder_extension) is used. Otherwise, the extension default to `'.d'`.

Mail::Box::Mbox->**foundIn**([$foldername], `%options`)
If no `$foldername` is specified, then the value of the `folder` option is taken. A mbox folder is a file which starts with a separator line: a line with `'From '` as first characters. Blank lines which start the file are ignored, which is not for all MUA’s acceptable.

```
 -Option             --Defined in     --Default
  folder                                undef
  folderdir            Mail::Box        undef
  subfolder_extension                   <from object>
```

folder => FOLDERNAME
folderdir => DIRECTORY
subfolder_extension => STRING

`$obj->`**lineSeparator**([<STRING|’CR’|’LF’|’CRLF’>])
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**locker**()
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**messageCreateOptions**([$type, `$config`])
Inherited, see ‘‘Internals’’ in Mail::Box::File

`$obj->`**moveAwaySubFolder**($directory, `$extension`)
Inherited, see ‘‘Internals’’ in Mail::Box::File

`$obj->`**parser**()
Inherited, see ‘‘Internals’’ in Mail::Box::File

`$obj->`**read**(%options)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**readMessages**(%options)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**storeMessage**($message)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**toBeThreaded**($messages)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**toBeUnthreaded**($messages)
Inherited, see ‘‘Internals’’ in Mail::Box

`$obj->`**updateMessages**(%options)
Inherited, see ‘‘Internals’’ in Mail::Box::File

`$obj->`**write**(%options)
Inherited, see ‘‘Internals’’ in Mail::Box::File

`$obj->`**writeMessages**(%options)
Inherited, see ‘‘Internals’’ in Mail::Box

Other methods
Extends ‘‘Other methods’’ in Mail::Box::File.

`$obj->`**timespan2seconds**($time)
Mail::Box::Mbox->**timespan2seconds**($time)
Inherited, see ‘‘Other methods’’ in Mail::Box

Error handling
Extends ‘‘Error handling’’ in Mail::Box::File.

`$obj->`**AUTOLOAD**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**addReport**($object)
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Mail::Box::Mbox->**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**errors**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**log**([$level, [$strings]])
Mail::Box::Mbox->**log**([$level, [$strings]])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**logPriority**($level)
Mail::Box::Mbox->**logPriority**($level)
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**logSettings**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**notImplemented**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**report**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**reportAll**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**trace**([$level])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**warnings**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

Cleanup
Extends ‘‘Cleanup’’ in Mail::Box::File.

`$obj->`**DESTROY**()
Inherited, see ‘‘Cleanup’’ in Mail::Box

DETAILS
Extends ‘‘DETAILS’’ in Mail::Box::File.

DETAILS
Extends ‘‘DETAILS’’ in Mail::Box::File.

Different kinds of folders
Extends ‘‘Different kinds of folders’’ in Mail::Box::File.

Available folder types
Extends ‘‘Available folder types’’ in Mail::Box::File.

Folder class implementation
Extends ‘‘Folder class implementation’’ in Mail::Box::File.

How MBOX folders work
MBOX folders store many messages in one file. Each message begins with a line which starts with the string `From ` . Lines inside a message which accidentally start with `From` are, in the file, preceded by ‘>’. This character is stripped when the message is read.

In this respect must be noted that the format of the MBOX files is not strictly defined. The exact content of the separator lines differ between Mail User Agents (MUA’s). Besides, some MUAs (like mutt) forget to encode the `From` lines within message bodies, breaking other parsers....

Simulation of sub-folders
MBOX folders do not have a sub-folder concept as directory based folders do, but this MBOX module tries to simulate them. In this implementation a directory like

```
Mail/subject1/
```

is taken as an empty folder `Mail/subject1`, with the folders in that directory as sub-folders for it. You may also use

```
Mail/subject1
Mail/subject1.d/
```

where `Mail/subject1` is the folder, and the folders in the `Mail/subject1.d` directory are used as sub-folders. If your situation is similar to the first example and you want to put messages in that empty folder, the directory is automatically (and transparently) renamed, so that the second situation is reached.

DIAGNOSTICS

Error: Cannot append messages to folder file `$filename:` $!
Appending messages to a not-opened file-organized folder may fail when the operating system does not allow write access to the file at hand.

Error: Cannot move away sub-folder `$dir`

Warning: Cannot remove folder `$name` file `$filename:` $!
Writing an empty folder will usually cause that folder to be removed, which fails for the indicated reason. new(remove_when_empty)

Warning: Cannot remove folder `$name` file `$filename:` $!
Writing an empty folder will usually cause that folder to be removed, which fails for the indicated reason. new(remove_when_empty) controls whether the empty folder will removed; setting it to false (`0`) may be needed to avoid this message.

Error: Cannot replace `$filename` by `$tempname`, to update folder `$name:` $!
The replace policy wrote a new folder file to update the existing, but was unable to give the final touch: replacing the old version of the folder file for the indicated reason.

Warning: Changes not written to read-only folder `$self`.
You have opened the folder read-only ––which is the default set by new(access)––, made modifications, and now want to close it. Set close(force) if you want to overrule the access mode, or close the folder with close(write) set to `NEVER`.

Error: Copying failed for one message.
For some reason, for instance disc full, removed by external process, or read-protection, it is impossible to copy one of the messages. Copying will proceed for the other messages.

Error: Destination folder `$name` is not writable.
The folder where the messages are copied to is not opened with write access (see new(access)). This has no relation with write permission to the folder which is controlled by your operating system.

Warning: Different messages with id `$msgid`
The message id is discovered more than once within the same folder, but the content of the message seems to be different. This should not be possible: each message must be unique.

Error: File too short to get write message `$nr` ($size, `$need`)
: Mail::Box is lazy: it tries to leave messages in the folders until they are used, which saves time and memory usage. When this message appears, something is terribly wrong: some lazy message are needed for updating the folder, but they cannot be retrieved from the original file anymore. In this case, messages can be lost.

  This message does appear regularly on Windows systems when using the 'replace' write policy. Please help to find the cause, probably something to do with Windows incorrectly handling multiple filehandles open in the same file.

Error: Folder `$name` not deleted: not writable.
: The folder must be opened with write access via new(access), otherwise removing it will be refused. So, you may have write-access according to the operating system, but that will not automatically mean that this `delete` method permits you to. The reverse remark is valid as well.

Error: Invalid timespan '$timespan' specified.
: The string does not follow the strict rules of the time span syntax which is permitted as parameter.

Warning: Message-id '$msgid' does not contain a domain.
: According to the RFCs, message-ids need to contain a unique random part, then an `@`, and then a domain name. This is made to avoid the creation of two messages with the same id. The warning emerges when the `@` is missing from the string.

Error: Package `$package` does not implement `$method`.
: Fatal error: the specific package (or one of its superclasses) does not implement this method where it should. This message means that some other related classes do implement this method however the class at hand does not. Probably you should investigate this and probably inform the author of the package.

Error: Unable to create subfolder `$name` of `$folder`.
: The copy includes the subfolders, but for some reason it was not possible to copy one of these. Copying will proceed for all other sub-folders.

Error: Unable to update folder `$self`.
: When a folder is to be written, both replace and inplace write policies are tried, If both fail, the whole update fails. You may see other, related, error messages to indicate the real problem.

SEE ALSO

This module is part of Mail-Box distribution version 3.009, built on August 18, 2020. Website: *http://perl.overmeer.net/CPAN/*

LICENSE