

NAME

Mail::Box::Maildir – handle Maildir folders

INHERITANCE

```
Mail::Box::Maildir
  is a Mail::Box::Dir
  is a Mail::Box
  is a Mail::Reporter
```

SYNOPSIS

```
use Mail::Box::Maildir;
my $folder = new Mail::Box::Maildir folder => $ENV{MAIL}, ...;
```

DESCRIPTION

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

Maildir is **not supported for Windows**, because it create filenames which are not accepted by the Windows system.

Extends “DESCRIPTION” in `Mail::Box::Dir`.

OVERLOADED

Extends “OVERLOADED” in `Mail::Box::Dir`.

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::Dir`.

Constructors

Extends “Constructors” in `Mail::Box::Dir`.

`Mail::Box::Maildir->new(%options)`

-Option	--Defined in	--Default
<code>accept_new</code>		<false>
<code>access</code>	<code>Mail::Box</code>	'r'
<code>body_delayed_type</code>	<code>Mail::Box</code>	<code>Mail::Message::Body::Delayed</code>
<code>body_type</code>	<code>Mail::Box</code>	<code>Mail::Message::Body::Lines</code>
<code>coerce_options</code>	<code>Mail::Box</code>	[]
<code>create</code>	<code>Mail::Box</code>	<false>
<code>directory</code>	<code>Mail::Box::Dir</code>	<derived from folder name>
<code>extract</code>	<code>Mail::Box</code>	10240
<code>field_type</code>	<code>Mail::Box</code>	undef
<code>fix_headers</code>	<code>Mail::Box</code>	<false>
<code>folder</code>	<code>Mail::Box</code>	<code>\$ENV{MAIL}</code>
<code>folderdir</code>	<code>Mail::Box</code>	<code>\$ENV{HOME}/.maildir</code>
<code>head_delayed_type</code>	<code>Mail::Box</code>	<code>Mail::Message::Head::Delayed</code>
<code>head_type</code>	<code>Mail::Box</code>	<code>Mail::Message::Head::Complete</code>
<code>keep_dups</code>	<code>Mail::Box</code>	<false>
<code>lock_file</code>	<code>Mail::Box</code>	<not used>
<code>lock_timeout</code>	<code>Mail::Box</code>	<not used>
<code>lock_type</code>	<code>Mail::Box</code>	'NONE' (constant)
<code>lock_wait</code>	<code>Mail::Box</code>	<not used>
<code>locker</code>	<code>Mail::Box</code>	undef

log	Mail::Reporter	'WARNINGS'
manager	Mail::Box	undef
message_type	Mail::Box	<folder-class>::Message
multipart_type	Mail::Box	Mail::Message::Body::Multipart
remove_when_empty	Mail::Box	<true>
save_on_exit	Mail::Box	<true>
trace	Mail::Reporter	'WARNINGS'
trusted	Mail::Box	<depends on folder location>

accept_new => BOOLEAN

When the folder is open, some messages may be stored in the new sub-directory. By default, these messages are immediately moved to the cur directory when the folder is opened. Otherwise, you have to call **acceptMessages()** or **Mail::Box::Maildir::Message::accept()**.

access => MODE

body_delayed_type => CLASS

body_type => CLASS|CODE

coerce_options => ARRAY

create => BOOLEAN

directory => DIRECTORY

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_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

trace => LEVEL

trusted => BOOLEAN

The folder

Extends “The folder” in Mail::Box::Dir.

\$obj->addMessage(\$message, %options)

Inherited, see “The folder” in Mail::Box

\$obj->addMessages(@messages)

Inherited, see “The folder” in Mail::Box

Mail::Box::Maildir->appendMessages(%options)

Inherited, see “The folder” in Mail::Box

\$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->directory()`
 Inherited, see “The folder” in Mail::Box::Dir

`$obj->folderdir([$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::Dir.

`$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::Dir.

`$obj->current([$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::Dir.

`$obj->listSubFolders(%options)`

Mail::Box::Maildir->**listSubFolders**(%options)

Inherited, see “Sub-folders” in Mail::Box

`$obj->nameOfSubFolder($subname, [$parentname])`

Mail::Box::Maildir->**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::Maildir->**topFolderWithMessages**()

Inherited, see “Sub-folders” in Mail::Box

Internals

Extends “Internals” in Mail::Box::Dir.

`$obj->acceptMessages()`

Accept all messages which are waiting in the new directory to be moved to the cur directory. This will not rescan the directory for newly arrived messages, because that’s a task for **update()**.

Mail::Box::Maildir->**appendMessage**(%options)

`$obj->coerce($message, %options)`

`$obj->create($foldername, %options)`

Mail::Box::Maildir->**create**(\$foldername, %options)

-Option --Defined in--Default

folderdir Mail::Box undef

folderdir => DIRECTORY

`$obj->createDirs($folderdir)`

Mail::Box::Maildir->**createDirs**(\$folderdir)

The \$folderdir contains the absolute path of the location where the messages are kept. Maildir folders contain a tmp, new, and cur sub-directory within that folder directory as well. This method will ensure that all directories exist. Returns false on failure.

`$obj->determineBodyType($message, $head)`

Inherited, see “Internals” in Mail::Box

`$obj->folderIsEmpty($folderdir)`

Mail::Box::Maildir->**folderIsEmpty**(\$folderdir)

Checks whether the folder whose directory is specified as absolute \$folderdir is empty or not. A folder is empty when the tmp, new, and cur subdirectories are empty and some files which are left there by application programs. The maildir spec explicitly states: .qmail, bulletintime, bulletinlock and seriallock. If any other files are found, the directory is considered not-empty.

`$obj->folderToDirectory($foldername, $folderdir)`

Inherited, see “Internals” in Mail::Box::Dir

`Mail::Box::Maildir->foundIn([$foldername], %options)`

Inherited, see “Internals” in Mail::Box

`$obj->lineSeparator([<STRING|'CR'|'LF'|'CRLF'>])`

Inherited, see “Internals” in Mail::Box

`$obj->locker()`

Inherited, see “Internals” in Mail::Box

`$obj->messageInFile($filename)`

Inherited, see “Internals” in Mail::Box::Dir

`$obj->read(%options)`

Inherited, see “Internals” in Mail::Box

`$obj->readMessageFileNames($directory)`

Inherited, see “Internals” in Mail::Box::Dir

`$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

`$obj->write(%options)`

Inherited, see “Internals” in Mail::Box

`$obj->writeMessages(%options)`

Inherited, see “Internals” in Mail::Box

Other methods

Extends “Other methods” in Mail::Box::Dir.

`$obj->timespan2seconds($time)`

`Mail::Box::Maildir->timespan2seconds($time)`

Inherited, see “Other methods” in Mail::Box

Error handling

Extends “Error handling” in Mail::Box::Dir.

`$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::Maildir->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::Maildir->log([\$level, [\$strings]])
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`
 Mail::Box::Maildir->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::Dir.

`$obj->DESTROY()`
 Inherited, see “Cleanup” in Mail::Box

DETAILS

The explanation is complicated, but for normal use you should bother yourself with all details.

Extends “DETAILS” in Mail::Box::Dir.

Different kinds of folders

Extends “Different kinds of folders” in Mail::Box::Dir.

Available folder types

Extends “Available folder types” in Mail::Box::Dir.

Folder class implementation

Extends “Folder class implementation” in Mail::Box::Dir.

How MAILDIR folders work

Maildir-type folders use a directory to store the messages of one folder. Each message is stored in a separate file. This seems useful, because changes in a folder change only a few of these small files, in contrast with file-based folders where changes in a folder cause rewrites of huge folder-files.

However, Maildir based folders perform very bad if you need header information of all messages. For instance, if you want to have full knowledge about all message-threads (see Mail::Box::Thread::Manager) in the folder, it requires to read all header lines in all message files. And usually, reading your messages as threads is desired. Maildir maintains a tiny amount of info visible in the filename, which may make it perform just a little bit faster than MH.

DIAGNOSTICS

Error: Cannot append Maildir message in \$new to folder \$self.

The message (or messages) could not be stored in the right directories for the Maildir folder.

Error: Cannot create Maildir directory \$dir: \$!

A Maildir folder is represented by a directory, with some sub-directories. The top folder directory could not be created for the reason indicated.

Error: Cannot create Maildir folder `$name`.

One or more of the directories required to administer a Maildir folder could not be created.

Error: Cannot create Maildir message file `$new`.

A message is converted from some other message format into a Maildir format by writing it to a file with a name which contains the status flags of the message. Apparently, creating this file failed.

Error: Cannot create Maildir subdir `$dir`: `!`

Each Maildir folder has three sub-directories for administration: `new`, `tmp`, and `cur`. The mentioned directory could not be created 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: Folder `$name` is opened read-only

You can not write to this folder unless you have opened the folder to write or append with `new(access)`, or the `force` option is set true.

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: Writing folder `$name` failed

For some reason (you probably got more error messages about this problem) it is impossible to write the folder, although you should because there were changes made.

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

Copyrights 2001–2020 by [Mark Overmeer]. For other contributors see ChangeLog.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.
See <http://dev.perl.org/licenses/>