

**NAME**

Mail::Box::MH::Message – one message in an MH–folder

**INHERITANCE**

```
Mail::Box::MH::Message
  is a Mail::Box::Dir::Message
  is a Mail::Box::Message
  is a Mail::Message
  is a Mail::Reporter
```

**SYNOPSIS**

```
my $folder = new Mail::Box::MH ...
my $message = $folder->message(10);
```

**DESCRIPTION**

A Mail::Box::MH::Message represents one message in an Mail::Box::MH folder . Each message is stored in a separate file, as for all Mail::Box::Dir folder types.

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

**METHODS**

Extends “METHODS” in Mail::Box::Dir::Message.

**Constructors**

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

`$obj->clone(%options)`

Inherited, see “Constructors” in Mail::Message

`Mail::Box::MH::Message->new(%options)`

Inherited, see “METHODS” in Mail::Box::Dir::Message

**Constructing a message**

Extends “Constructing a message” in Mail::Box::Dir::Message.

`$obj->bounce( [<$rg_object|%options>] )`

Inherited, see “Constructing a message” in Mail::Message::Construct::Bounce

`Mail::Box::MH::Message->build( [$message|$part|$body], $content )`

Inherited, see “Constructing a message” in Mail::Message::Construct::Build

`Mail::Box::MH::Message->buildFromBody($body, [$head], $headers)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Build

`$obj->forward(%options)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardAttach(%options)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardEncapsulate(%options)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardInline(%options)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardNo(%options)`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardPostlude()`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardPrelude()`

Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

`$obj->forwardSubject(String)`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Forward

Mail::Box::MH::Message->**read**(\$fh|String|SCALAR|ARRAY, %options)  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Read

`$obj->rebuild(%options)`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Rebuild

`$obj->reply(%options)`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Reply

`$obj->replyPrelude( [String|$field|$address|ARRAY-$of-$things] )`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Reply

`$obj->replySubject(String)`  
 Mail::Box::MH::Message->**replySubject**(String)  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Reply

### The message

Extends “The message” in Mail::Box::Dir::Message.

`$obj->container()`  
 Inherited, see “The message” in Mail::Message

`$obj->copyTo($folder, %options)`  
 Inherited, see “The message” in Mail::Box::Message

`$obj->filename( [$filename] )`  
 Inherited, see “The message” in Mail::Box::Dir::Message

`$obj->folder( [$folder] )`  
 Inherited, see “The message” in Mail::Box::Message

`$obj->isDummy()`  
 Inherited, see “The message” in Mail::Message

`$obj->isPart()`  
 Inherited, see “The message” in Mail::Message

`$obj->messageId()`  
 Inherited, see “The message” in Mail::Message

`$obj->moveTo($folder, %options)`  
 Inherited, see “The message” in Mail::Box::Message

`$obj->partNumber()`  
 Inherited, see “The message” in Mail::Message

`$obj->print( [$fh] )`  
 Inherited, see “The message” in Mail::Message

`$obj->send( [$mailer], %options )`  
 Inherited, see “The message” in Mail::Message

`$obj->seqnr( [$integer] )`  
 The order of this message in the folder, counted from zero. Do not change the number (unless you understand the implications).

WARNING: This sequence number has nothing to do with the message’s filename, which in case of MH folders are also numbers! If you need that one, use the File::Basename subroutine `basename` of the filename.

`$obj->size()`  
 Inherited, see “The message” in Mail::Message

`$obj->toptlevel()`  
 Inherited, see “The message” in Mail::Message

`$obj->write( [$fh] )`  
 Inherited, see “The message” in Mail::Message

### The header

Extends “The header” in Mail::Box::Dir::Message.

`$obj->bcc()`  
 Inherited, see “The header” in Mail::Message

`$obj->cc()`  
 Inherited, see “The header” in Mail::Message

`$obj->date()`  
 Inherited, see “The header” in Mail::Message

`$obj->destinations()`  
 Inherited, see “The header” in Mail::Message

`$obj->from()`  
 Inherited, see “The header” in Mail::Message

`$obj->get($fieldname)`  
 Inherited, see “The header” in Mail::Message

`$obj->guessTimestamp()`  
 Inherited, see “The header” in Mail::Message

`$obj->head( [$head] )`  
 Inherited, see “The header” in Mail::Message

`$obj->nrLines()`  
 Inherited, see “The header” in Mail::Message

`$obj->sender()`  
 Inherited, see “The header” in Mail::Message

`$obj->study($fieldname)`  
 Inherited, see “The header” in Mail::Message

`$obj->subject()`  
 Inherited, see “The header” in Mail::Message

`$obj->timestamp()`  
 Inherited, see “The header” in Mail::Message

`$obj->to()`  
 Inherited, see “The header” in Mail::Message

### The body

Extends “The body” in Mail::Box::Dir::Message.

`$obj->body( [$body] )`  
 Inherited, see “The body” in Mail::Message

`$obj->contentType()`  
 Inherited, see “The body” in Mail::Message

`$obj->decoded(%options)`  
 Inherited, see “The body” in Mail::Message

`$obj->encode(%options)`  
 Inherited, see “The body” in Mail::Message

`$obj->isMultipart()`

Inherited, see “The body” in Mail::Message

`$obj->isNested()`

Inherited, see “The body” in Mail::Message

`$obj->parts( [<'ALL'|'ACTIVE'|'DELETED'|'RECURSE'|$filter>] )`

Inherited, see “The body” in Mail::Message

## Flags

Extends “Flags” in Mail::Box::Dir::Message.

`$obj->delete()`

Inherited, see “Flags” in Mail::Message

`$obj->deleted( [BOOLEAN] )`

Inherited, see “Flags” in Mail::Message

`$obj->isDeleted()`

Inherited, see “Flags” in Mail::Message

`$obj->isModified()`

Inherited, see “Flags” in Mail::Message

`$obj->label($label|PAIRS)`

Inherited, see “Flags” in Mail::Message

`$obj->labels()`

Inherited, see “Flags” in Mail::Message

`$obj->labelsToStatus()`

Inherited, see “Flags” in Mail::Message

`$obj->modified( [BOOLEAN] )`

Inherited, see “Flags” in Mail::Message

`$obj->statusToLabels()`

Inherited, see “Flags” in Mail::Message

## The whole message as text

Extends “The whole message as text” in Mail::Box::Dir::Message.

`$obj->file()`

Inherited, see “The whole message as text” in Mail::Message::Construct::Text

`$obj->lines()`

Inherited, see “The whole message as text” in Mail::Message::Construct::Text

`$obj->printStructure( [$fh|undef],[ $indent] )`

Inherited, see “The whole message as text” in Mail::Message::Construct::Text

`$obj->string()`

Inherited, see “The whole message as text” in Mail::Message::Construct::Text

## Internals

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

`$obj->clonedFrom()`

Inherited, see “Internals” in Mail::Message

`Mail::Box::MH::Message->coerce($message, %options)`

Inherited, see “Internals” in Mail::Message

`$obj->create($filename)`

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

`$obj->diskDelete()`  
 Inherited, see “Internals” in Mail::Box::Message

`$obj->isDelayed()`  
 Inherited, see “Internals” in Mail::Message

`$obj->loadBody()`  
 Inherited, see “Internals” in Mail::Box::Dir::Message

`$obj->loadHead()`  
 Inherited, see “Internals” in Mail::Box::Dir::Message

`$obj->parser()`  
 Inherited, see “Internals” in Mail::Box::Dir::Message

`$obj->readBody( $parser, $head, [$bodytype] )`  
 Inherited, see “Internals” in Mail::Box::Message

`$obj->readFromParser( $parser, [$bodytype] )`  
 Inherited, see “Internals” in Mail::Message

`$obj->readHead( $parser, [$class] )`  
 Inherited, see “Internals” in Mail::Message

`$obj->recursiveRebuildPart($part, %options)`  
 Inherited, see “Internals” in Mail::Message::Construct::Rebuild

`$obj->storeBody($body)`  
 Inherited, see “Internals” in Mail::Message

`$obj->takeMessageId( [STRING] )`  
 Inherited, see “Internals” in Mail::Message

### Error handling

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

`$obj->AUTOLOAD()`  
 Inherited, see “METHODS” in Mail::Message::Construct

`$obj->addReport($object)`  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->defaultTrace( [$level][[$loglevel, $tracelevel]][[$level, $callback] ]`  
`Mail::Box::MH::Message->defaultTrace( [$level][[$loglevel, $tracelevel]][[$level, $callback] ]`  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->errors()`  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->log( [$level, [$strings]] )`  
`Mail::Box::MH::Message->log( [$level, [$strings]] )`  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`  
`Mail::Box::MH::Message->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

`$obj->shortSize( [$value] )`  
 Mail::Box::MH::Message->shortSize( [\$value] )  
 Inherited, see “Error handling” in Mail::Message

`$obj->shortString()`  
 Inherited, see “Error handling” in Mail::Message

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

`$obj->warnings()`  
 Inherited, see “Error handling” in Mail::Reporter

**Cleanup**

Extends “Cleanup” in Mail::Box::Dir::Message.

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

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

**DETAILS**

Extends “DETAILS” in Mail::Box::Dir::Message.

**Structure of a Message**

Extends “Structure of a Message” in Mail::Box::Dir::Message.

**Message object implementation**

Extends “Message object implementation” in Mail::Box::Dir::Message.

**Message class implementation**

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

**Labels**

Extends “Labels” in Mail::Box::Dir::Message.

*Predefined labels*

Extends “Predefined labels” in Mail::Box::Dir::Message.

*Status and X-Status fields*

Extends “Status and X-Status fields” in Mail::Box::Dir::Message.

*The .mh\_sequences file*

Typically, the file which contains the labels is called `.mh_sequences`. The MH messages are numbered from 1. As example content for `.mh_sequences`:

```
cur: 93
unseen: 32 35-56 67-80
```

To generalize labels on messages, two are treated specially:

- `cur`  
 The `cur` specifies the number of the message where the user stopped reading mail from this folder at last access. Internally in these modules referred to as label `current`.
- `unseen`  
 With `unseen` is listed which message was never read. This must be a mistake in the design of MH: it must be a source of confusion. People should never use labels with a negation in the name:

```

    if($seen)           if(!$unseen)    #yuk!
    if(!$seen)          if($unseen)
    unless($seen)       unless($unseen) #yuk!

```

So: label unseen is translated into seen for internal use.

## DIAGNOSTICS

Error: Cannot coerce a `$class` object into a `$class` object

Error: Cannot create parser for `$filename`.

For some reason (the previous message have told you already) it was not possible to create a message parser for the specified filename.

Error: Cannot include forward source as `$include`.

Unknown alternative for the forward(include). Valid choices are NO, INLINE, ATTACH, and ENCAPSULATE.

Error: Cannot include reply source as `$include`.

Unknown alternative for the include option of **reply()**. Valid choices are NO, INLINE, and ATTACH.

Error: Cannot write message to `$filename: $!`

When a modified or new message is written to disk, it is first written to a temporary file in the folder directory. For some reason, it is impossible to create this file.

Error: Failed to move `$new` to `$filename: $!`

When a modified or new message is written to disk, it is first written to a temporary file in the folder directory. Then, the new file is moved to replace the existing file. Apparently, the latter fails.

Error: Method bounce requires To, Cc, or Bcc

The message **bounce()** method forwards a received message off to someone else without modification; you must specified it's new destination. If you have the urge not to specify any destination, you probably are looking for **reply()**. When you wish to modify the content, use **forward()**.

Error: Method forwardAttach requires a preamble

Error: Method forwardEncapsulate requires a preamble

Error: No address to create forwarded to.

If a forward message is created, a destination address must be specified.

Error: No default mailer found to send message.

The message **send()** mechanism had not enough information to automatically find a mail transfer agent to sent this message. Specify a mailer explicitly using the `via` options.

Error: No rebuild rule `$name` defined.

Error: Only **build()** Mail::Message's; they are not in a folder yet

You may wish to construct a message to be stored in a some kind of folder, but you need to do that in two steps. First, create a normal Mail::Message, and then add it to the folder. During this **Mail::Box::addMessage()** process, the message will get **coerce()**-d into the right message type, adding storage information and the like.

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 read delayed body.

For some reason, the header of the message could be read, but the body cannot. Probably the file has disappeared or the permissions were changed during the progress of the program.

Error: Unable to read delayed head.

Mail::Box tries to be *lazy* with respect to parsing messages. When a directory organized folder is opened, only the filenames of messages are collected. At first use, the messages are read from their

file. Apparently, a message is used for the first time here, but has disappeared or is unreadable for some other reason.

Error: coercion starts with some object

## 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/>