NAME

Mail::Box::Message − manage one message within a mail−folder

INHERITANCE

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

Mail::Box::Message is extended by
  Mail::Box::Dir::Message
  Mail::Box::File::Message
  Mail::Box::Message::Destructed
  Mail::Box::Net::Message
```

SYNOPSIS

```
# Usually these message objects are created indirectly
use Mail::Box::Manager;
my $manager = Mail::Box::Manager->new;
my $folder  = $manager->open(folder => 'Mail/Drafts');
my $msg     = $folder->message(1);
$msg->delete;
$msg->size;   # and much more
```

DESCRIPTION

These pages do only describe methods which relate to folders. If you access the knowledge of a message, then read Mail::Message.

During its life, a message will pass through certain stages. These stages were introduced to reduce the access-time to the folder. Changing from stage, the message's body and head objects may change.

Extends ''DESCRIPTION'' in Mail::Message.

METHODS

Extends ''METHODS'' in Mail::Message.

Constructors

Extends ''Constructors'' in Mail::Message.

$obj->**clone**(%options)

Inherited, see ''Constructors'' in Mail::Message

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

```
 -Option    --Defined in     --Default
  body        Mail::Message    undef
  body_type                    <from folder>
  deleted     Mail::Message    <false>
  field_type  Mail::Message    undef
  folder                       <required>
  head        Mail::Message    undef
  head_type   Mail::Message    Mail::Message::Head::Complete
  labels      Mail::Message    {}
  log         Mail::Reporter   'WARNINGS'
  messageId   Mail::Message    undef
  modified    Mail::Message    <false>
  size                         undef
  trace       Mail::Reporter   'WARNINGS'
  trusted     Mail::Message    <false>
```

body => OBJECT
body_type => CODE|CLASS
If the body of a message is used delay-loaded, the message must what type of message to become when it finally gets parsed. The folder which is delaying the load must specify the algorithm to determine that type.

deleted => BOOLEAN
field_type => CLASS
folder => FOLDER
The folder where this message appeared in. The argument is an instance of (a sub-class of) a Mail::Box.

head => OBJECT
head_type => CLASS
labels => ARRAY|HASH
log => LEVEL
messageId => STRING
modified => BOOLEAN
size => INTEGER
The size of the message, which includes head and body, but without the message separators which may be used by the folder type.

trace => LEVEL
trusted => BOOLEAN

Constructing a message

Extends ‘‘Constructing a message’’ in Mail::Message.

`$obj`−>**bounce**([<$rg_object|%options>])
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Bounce

Mail::Box::Message−>**build**([$message|$part|$body], `$content`)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Build

Mail::Box::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::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::Message->**replySubject**(STRING)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Reply

The message
Extends ‘‘The message’’ in Mail::Message.

`$obj->`**container**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**copyTo**($folder, `%options`)
Copy the message to the indicated opened `$folder`, without deleting the original. The coerced message (the clone in the destination folder) is returned.

```
-Option      --Default
 shallow       <false>
 shallow_body  <false>
 shallow_head  <false>
 share         <false>
```

shallow => BOOLEAN
Used for clone(shallow).

shallow_body => BOOLEAN
Used for clone(shallow_body).

shallow_head => BOOLEAN
Used for clone(shallow_head).

share => BOOLEAN
Try to share the physical storage of the message between the two folders. Sometimes, they even may be of different types. When not possible, this options will be silently ignored.

example:

```
my $draft = $mgr->open(folder => 'Draft');
$message->copyTo($draft, share => 1);
```

`$obj->`**folder**([$folder])
In with folder did we detect this message/dummy? This is a reference to the folder-object.

`$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`)
Move the message from this folder to the `$folder` specified. This will create a copy using **clone()** first. Then, this original message is flagged to get deleted. So until the source folder is closed, two copies of the message may stay in memory.

The newly created message clone (part of the destination folder) is returned. All `%options` are

passed to **copyTo()**

```
-Option      --Default
 shallow_body  <undef>
 share         <true unless shallow_body exists>
```

shallow_body => BOOLEAN
Only create a shallow body, which means that the header can not be reused. A message can therefore not be shared in storage unless explicitly stated.

share => BOOLEAN
When there is a chance that the original message can be undeleted, then this must be set to false. Otherwise a shallow clone will be made, which will share the header which can be modified in the undeleted message.

example: of moving a message

```
my $trash = Mail::Box::Mbox->new(folder => 'trash');
my $t = $msg->moveTo($trash);
```

is equivalent to

```
my $t = $msg->copyTo($trash, share => 1);
$msg->delete;
```

`$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])
Get the number of this message is the current folder. It starts counting from zero. Do not change the number.

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

`$obj->`**toplevel**()
Inherited, see ‘‘The message’’ in Mail::Message

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

The header
Extends ‘‘The header’’ in Mail::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::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::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::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::Message.

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

Mail::Box::Message->**coerce**($message, `%options`)
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj`->**diskDelete**()
Remove a message from disk. This is not from the folder, but everything else, like parts of the message which are stored outside from the folder.

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

`$obj`->**readBody**(`$parser`, `$head`, [$bodytype])
Read the body of one message. The `$parser` gives access to the folder file. The `$head` has been read with **readHead()**. The optional`$bodytype` supplies the class name of the body to be created, or a code reference to a routine which can produce a body type based on the head (passed as first argument).

By default, the `$bodytype` will call **Mail::Box::determineBodyType()** where the message will be added to.

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

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

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

`$obj->`**destruct**()
Removes most of the memory occupied by the message by detaching the header and body. Then, the object changes into a Mail::Box::Message::Destructed which will catch all attempts to access the header and body. Be careful with the usage of this method.

DETAILS

Extends ‘‘DETAILS’’ in Mail::Message.

DIAGNOSTICS

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

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: 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: 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