

**NAME**

Mail::Message::Replace::MailInternet – fake Mail::Internet

**INHERITANCE**

```
Mail::Message::Replace::MailInternet
    is a Mail::Message
    is a Mail::Reporter
```

**SYNOPSIS**

```
!!! BETA !!!

# change
use Mail::Internet;
# into
use Mail::Message::Replace::MailInternet;
# in existing code, and the code should still work, but
# with the Mail::Message features.
```

**DESCRIPTION**

This module is a wrapper around a Mail::Message, which simulates a Mail::Internet object. The namespace of that module is hijacked and many methods are added.

Most methods will work without any change, but you may need to have a look at your **smtpsend()** and **send()** calls.

Extends “DESCRIPTION” in Mail::Message.

**OVERLOADED****METHODS**

Extends “METHODS” in Mail::Message.

**Constructors**

Extends “Constructors” in Mail::Message.

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

Inherited, see “Constructors” in Mail::Message

**\$obj->dup()**

Duplicate the message. The result will again be a Mail::Internet compatible object.

**\$obj->empty()**

Remove all data from this object. Very dangerous!

**Mail::Message::Replace::MailInternet->new( [\$arg], [%options] )**

-Option	--Defined in	--Default
Body		undef
FoldLength		79
Header		undef
MailFrom		'KEEP'
Modify		0
body	Mail::Message	undef
body_type	Mail::Message	Mail::Message::Body::Lines
deleted	Mail::Message	<false>
field_type	Mail::Message	undef
head	Mail::Message	undef
head_type	Mail::Message	Mail::Message::Replace::MailHeader
labels	Mail::Message	{ }
log	Mail::Reporter	'WARNINGS'
messageId	Mail::Message	undef
modified	Mail::Message	<false>
trace	Mail::Reporter	'WARNINGS'

```

    trusted      Mail::Message      <false>

Body => \@lines
    Array of "\n" terminated lines. If not specified, the lines will be read from $arg.

FoldLength => INTEGER
    Number of characters permitted on any refolded header line. Passed to
    Mail::Message::Replace::MailHeader::new(FoldLength).

Header => OBJECT
    The Mail::Header object, which is passed here, is a fake one as well... It is translated into a
    new(head). If not given, the header will be parsed from the $arg.

MailFrom => 'IGNORE'|'ERROR'|'COERCE'|'KEEP'
    What to do with leading "From " lines in e-mail data. Passed to
    Mail::Message::Replace::MailHeader::new(MailFrom).

Modify => BOOLEAN
    Whether to re-fold all the incoming fields. Passed to
    Mail::Message::Replace::MailHeader::new(Modify).

body => OBJECT
body_type => CLASS
deleted => BOOLEAN
field_type => CLASS
head => OBJECT
head_type => CLASS
labels => ARRAY|HASH
log => LEVEL
messageId => STRING
modified => BOOLEAN
trace => LEVEL
trusted => BOOLEAN

```

example: replace traditional Mail::Internet by this wrapper

```

# was
use Mail::Internet;
my $mi = Mail::Internet->new(@options);

# becomes
use Mail::Message::Replace::MailInternet;
my $mi = Mail::Internet->new(@options);

```

### Attributes

`$obj->MailFrom( [STRING] )`  
 Your email address.

### Constructing a message

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

`$obj->add_signature( [$filename] )`  
 Replaced by **sign()**, but still usable. `$filename` is the file which contains the signature, which defaults to `$ENV{HOME}/.signature`.

`$obj->bounce( [<$rg_object%options>] )`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Bounce

`Mail::Message::Replace::MailInternet->build( [$message|$part|$body], $content )`  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Build

**Mail::Message::Replace::MailInternet**→**buildFromBody**(\$body, [\$head], \$headers)  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Build

**\$obj**→**extract**(\@lines|\$fh)  
 Read header and body from an ARRAY or \$fh

**\$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

**\$obj**→**read**(\@lines|\$fh, %options)  
**Mail::Message::Replace::MailInternet**→**read**(\@lines|\$fh, %options)  
 Read header and body from the specified ARRAY or \$fh. When used as object method, **Mail::Message::read()** is called, to be MailBox compliant. As class method, the Mail::Internet compatible read is called. %options are only available in the first case.

-Option	--Defined in	--Default
body_type	Mail::Message::Construct::Read	undef
strip_status_fields	Mail::Message::Construct::Read	<true>

body\_type => CLASS  
 strip\_status\_fields => BOOLEAN

**\$obj**→**read\_body**(\@lines|\$fh)  
 Read only the message’s body from the ARRAY or \$fh.

**\$obj**→**read\_header**(\@lines|\$fh)  
 Read only the message’s header from the ARRAY or \$fh

**\$obj**→**rebuild**(%options)  
 Inherited, see “Constructing a message” in Mail::Message::Construct::Rebuild

**\$obj**→**reply**(%options)  
 BE WARNED: the main job for creating a reply is done by **Mail::Message::reply()**, which may produce a result which is compatible, but may be different from Mail::Internet’s version.

-Option	--Defined in	--Default
Bcc	Mail::Message::Construct::Reply	undef
Cc	Mail::Message::Construct::Reply	<'cc' in current>
Exclude		[]
From	Mail::Message::Construct::Reply	<'to' in current>
Inline		>
Keep		[]
Message-ID	Mail::Message::Construct::Reply	<uniquely generated>
ReplyAll		<false>
Subject	Mail::Message::Construct::Reply	replySubject()
To	Mail::Message::Construct::Reply	<sender in current>
body	Mail::Message::Construct::Reply	undef
group_reply	Mail::Message::Construct::Reply	<true>
header_template		\$ENV{HOME}/.mailhdr
include	Mail::Message::Construct::Reply	'INLINE'
max_signature	Mail::Message::Construct::Reply	10
message_type	Mail::Message::Construct::Reply	Mail::Message
postlude	Mail::Message::Construct::Reply	undef
prelude	Mail::Message::Construct::Reply	undef
quote	Mail::Message::Construct::Reply	'> '
signature	Mail::Message::Construct::Reply	undef
strip_signature	Mail::Message::Construct::Reply	qr/^--\s/

Bcc => ADDRESSES

Cc => ADDRESSES

Exclude => \@names

Remove the fields with the specified names from the produced reply message.

From => ADDRESSES

Inline => STRING

Quotation STRING, which is translated into reply(quote). The normal default of quote is "> ", in stead of ">".

Keep => \@names

Copy all header fields with the specified NAMES from the source to the reply message.

Message-ID => STRING

ReplyAll => BOOLEAN

Reply to the group? Translated into reply(group\_reply), which has as default the exact oposite of this option, being true.

Subject => STRING|CODE

To => ADDRESSES

body => BODY

group\_reply => BOOLEAN

header\_template => \$filename|undef

Read the return header from the template file. When this is explicitly set to undef, or the file does not exist, then a header will be created.

include => 'NO'|'INLINE'|'ATTACH'

max\_signature => INTEGER

message\_type => CLASS

postlude => BODY|LINES

prelude => BODY|LINES

quote => CODE|STRING

signature => BODY|MESSAGE

```

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

$obj->replySubject(STRING)
Mail::Message::Replace::MailInternet->replySubject(STRING)
    Inherited, see “Constructing a message” in Mail::Message::Construct::Reply

$obj->sign(%options)
    Add a signature (a few extra lines) to the message.

    -Option      --Default
    File         undef
    Signature    ''

File => $filename
    Specifies a filename where the signature is in.

Signature => STRING\|@lines
    The signature in memory.

```

### The message

```

Extends “The message” in Mail::Message.

$obj->container()
    Inherited, see “The message” in Mail::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->nntppost(%options)
    Send an NNTP message (newsgroup message), which is equivalent to Mail::Transport::NNTP or
    Mail::Message::send() with via 'nntp'.

    -Option--Default
    Debug    <false>
    Host     <from Net::Config>
    Port     119

Debug => BOOLEAN
Host => $hostname
Port => INTEGER

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

$obj->print( [$fh] )
    Prints the whole message to the specified $fh, which default to STDOUT. This calls
    Mail::Message::print().

$obj->send($type, %options)
    Send via Mail Transfer Agents (MUA). These will be handled by various Mail::Transport::Send
    extensions. The test $type is not supported.

$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::Message.

`$obj->add(\@lines)`  
Add header lines, which simply calls `Mail::Message::Header::add()` on the header for each specified LINE. The last added LINE is returned.

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

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

`$obj->clean_header()`  
Not to be used, replaced by `header()`.

`$obj->combine( $tag, [$with] )`  
Not implemented, because I see no use for it.

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

`$obj->delete( $name, [$index] )`  
Delete the fields with the specified \$name. The deleted fields are returned.  
  
BE WARNED: if no \$name is specified, the `delete` is interpreted as the deletion of the message in a folder, so `Mail::Box::Message::delete()` will be called. This may have no negative effect at all...

Calls `Mail::Message::Replace::MailHeader::delete()`

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

`$obj->fold( [$length] )`  
Fold all the fields to a certain maximum \$length. Implemented by `Mail::Message::Replace::MailHeader::fold()`

`$obj->fold_length( [$tag, $length] )`  
Set the maximum line \$length. \$tag is ignored. Implemented by `Mail::Message::Replace::MailHeader::fold_length()`

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

`$obj->get( $name, [$index] )`  
Get all the header fields with the specified \$name. In scalar context, only the first fitting \$name is returned. Even when only one \$name is specified, multiple lines may be returned: some fields appear more than once in a header. Calls `Mail::Message::Replace::MailHeader::get()`

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

`$obj->head( [$head] )`  
Returns the head of the message, or creates an empty one if none is defined. The \$head argument, which sets the header, is not available for Mail::Internet, but is there to be compatible with the head method of Mail::Message.

`$obj->header( \@lines )`  
Optionally reads a header from the ARRAY of lines, and then returns those fields as array-ref nicely folded. Implemented by `Mail::Message::Replace::MailHeader::header()`

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

`$obj->print_header($fh)`  
 Calls **Mail::Message::Head::Complete::print()**.

`$obj->replace( $tag, $line, [$index] )`  
 Adds LINES to the header, but removes fields with the same name if they already exist. Calls **Mail::Message::Replace::MailHeader::replace()**

`$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->tidy_headers()`  
 No effect anymore (always performed).

`$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.

`$obj->body($lines|@lines)`  
 Returns an ARRAY of lines, representing the body. With arguments, a new body will be created. In Mail::Internet, the body is not an object but a simple array.

BE WARNED: this overrules the **Mail::Message::body()** method, which may cause some confusion. Use **bodyObject()** to get access to that body’s data.

`$obj->bodyObject( [$body] )`  
 Calls **Mail::Message::body()**, because that body method is overruled by the one which has a Mail::Internet compatible interface.

`$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

`$obj->print_body( [$fh] )`  
 Prints the body to the specified `$fh`, which defaults to STDOUT. This calls **Mail::Message::Body::print()**.

`$obj->remove_sig( [$nrlines] )`

Remove the signature of a message with a maximum of `$nrlines` lines, which defaults to 10. The work is done on the decoded body content, by **Mail::Message::Body::stripSignature()**.

`$obj->smtpsend(%options)`

This method is calling **Mail::Message::send()** via `smtp`, which is implemented in `Mail::Transport::SMTP`. The implementation is slightly different, so this method is not 100% compliant.

```
-Option  --Default
Debug      <false>
Hello      <helo_domain from Net::Config>
Host       $ENV{SMTPHOSTS} or from Net::Config
MailFrom   $ENV{MAILADDRESS} or $ENV{USER}
Port       25
```

Debug => BOOLEAN

Hello => STRING

Host => HOSTNAME

Only the first detected HOSTNAME is taken, so differs from the original implementation.

MailFrom => STRING

Your e-mail address. This simulated `Mail::Internet` object does not try to create an e-mail address from the `sendmail` configuration file, because that is generally a bad idea in environments with virtual hosts, as we have now-adays.

Port => INTEGER

`$obj->tidy_body()`

Removes blank lines from begin and end of the body.

## Flags

Extends “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->as_mbox_string()`

Returns the whole message as one string, which can be included in an MBOX folder (while not using `Mail::Box::Mbox`). Lines in the body which start with `From` are escaped with an `>`.



`$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

### The nasty bits

`$obj->isa($class)`  
`Mail::Message::Replace::MailInternet->isa($class)`  
 Of course, the `isa( )` class inheritance check should not see our nasty trick.

### Internals

Extends “Internals” in Mail::Message.

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

`Mail::Message::Replace::MailInternet->coerce($message)`  
 Coerce (adapt type) of the specified `$message` (anything **Mail::Message::coerce()** accepts) into an Mail::Internet simulating object.

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

`$obj->readBody( $parser, $head, [$bodytype] )`  
 Inherited, see “Internals” in Mail::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::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, $callback] ]`  
`Mail::Message::Replace::MailInternet->defaultTrace( [$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::Message::Replace::MailInternet->log( [\$level, [\$strings]] )  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`  
 Mail::Message::Replace::MailInternet->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::Message::Replace::MailInternet->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()`  
 Inherited, see “Cleanup” in Mail::Message

## DETAILS

Extends “DETAILS” in Mail::Message.

## DIAGNOSTICS

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

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

Error: Mail::Internet does not support this kind of data

The ARGV data can only be a file handle or an ARRAY. Other data types are not supported (see `read()` if you want to have more).

Error: Method bounce requires To, Cc, or Bcc

The message `bounce()` method forwards a received message off to someone else without modification; you must specify its 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 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.

## SEE ALSO

This module is part of Mail-Message distribution version 3.012, built on February 11, 2022. Website: <http://perl.overmeer.net/CPAN/>

## LICENSE

Copyrights 2001–2022 by [Mark Overmeer <markov@cpan.org>]. 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/>