

NAME

Mail::Message::Replace::MailHeader – fake Mail::Header

INHERITANCE

```
Mail::Message::Replace::MailHeader
  is a Mail::Message::Head::Complete
  is a Mail::Message::Head
  is a Mail::Reporter
```

SYNOPSIS

```
!!! ALPHA CODE !!!

# change
use Mail::Internet;
use Mail::Header;
# 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::Head::Complete, which simulates a Mail::Header object. The name-space of that module is hijacked and many methods are added.

Most methods will work without any change, but you should test your software again. Small changes have been made to **fold_length()**, **header_hashref()**.

Extends “DESCRIPTION” in Mail::Message::Head::Complete.

OVERLOADED

Extends “OVERLOADED” in Mail::Message::Head::Complete.

overload: “”

Inherited, see “OVERLOADED” in Mail::Message::Head

overload: **bool**

Inherited, see “OVERLOADED” in Mail::Message::Head

METHODS

Extends “METHODS” in Mail::Message::Head::Complete.

Constructors

Extends “Constructors” in Mail::Message::Head::Complete.

Mail::Message::Replace::MailHeader->**build**([PAIR|\$field], ...)

Inherited, see “Constructors” in Mail::Message::Head::Complete

\$obj->**clone**([@names|ARRAY|Regexps])

Inherited, see “Constructors” in Mail::Message::Head::Complete

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

The \$arg is an array with header lines.

Option	--Defined in	--Default
FoldLength		79
MailFrom		'KEEP'
Modify		<false>
field_type	Mail::Message::Head	Mail::Message::Field::Fast
log	Mail::Reporter	'WARNINGS'
message	Mail::Message::Head	undef
modified	Mail::Message::Head	<false>
trace	Mail::Reporter	'WARNINGS'

FoldLength => INTEGER
 MailFrom => 'IGNORE'|'ERROR'|'COERCE'|'KEEP'
 How to handle the From lines. See **mail_from()**.
 Modify => BOOLEAN
 Reformat all header lines when they come in: change the folding.
 field_type => CLASS
 log => LEVEL
 message => MESSAGE
 modified => BOOLEAN
 trace => LEVEL

The header

Extends “The header” in Mail::Message::Head::Complete.

\$obj->isDelayed()
 Inherited, see “The header” in Mail::Message::Head
\$obj->isEmpty()
 Inherited, see “The header” in Mail::Message::Head
\$obj->isModified()
 Inherited, see “The header” in Mail::Message::Head
\$obj->knownNames()
 Inherited, see “The header” in Mail::Message::Head
\$obj->message([\$message])
 Inherited, see “The header” in Mail::Message::Head
\$obj->modified([BOOLEAN])
 Inherited, see “The header” in Mail::Message::Head
\$obj->nrLines()
 Inherited, see “The header” in Mail::Message::Head::Complete
\$obj->orderedFields()
 Inherited, see “The header” in Mail::Message::Head
\$obj->size()
 Inherited, see “The header” in Mail::Message::Head::Complete
\$obj->wrap(\$integer)
 Inherited, see “The header” in Mail::Message::Head::Complete

Access to the header

Extends “Access to the header” in Mail::Message::Head::Complete.

\$obj->add(\$line, [\$index])
 Add a header line, which simply calls Mail::Message::Head::add() on the header for the specified \$line. The \$index is ignored, the unfolded body of the field is returned.
\$obj->addListGroup(\$object)
 Inherited, see “Access to the header” in Mail::Message::Head::Complete
\$obj->addResentGroup(\$resent_group|\$data)
 Inherited, see “Access to the header” in Mail::Message::Head::Complete
\$obj->addSpamGroup(\$object)
 Inherited, see “Access to the header” in Mail::Message::Head::Complete
\$obj->count(\$name)
 Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->delete(\$tag, [\$index])
Delete the fields with the specified \$tag. The deleted fields are returned. If no index is given, then all are removed.

\$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 in list context: some fields appear more than once in a header.

\$obj->grepNames([@names|ARRAY|Regexps])
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->listGroup()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->names()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->print([\$fh])
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->printSelected(\$fh, <STRING|Regex>, ...)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->printUndisclosed([\$fh])
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeContentInfo()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeField(\$field)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeFields(<STRING|Regex>, ...)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeFieldsExcept(<STRING|Regex>, ...)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeListGroup()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeResentGroups()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->removeSpamGroups()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->replace(\$tag, \$line, [\$index])
Replace the field named \$tag. from place \$index (by default the first) by the \$line. When \$tag is undef, it will be extracted from the \$line first. This calls **Mail::Message::Head::Complete::reset()** on the message’s head.

\$obj->resentGroups()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->reset(\$name, @fields)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->set(\$field| \$line| <\$name, \$body, [\$attrs]>)
Inherited, see “Access to the header” in Mail::Message::Head::Complete

\$obj->spamDetected()
Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->spamGroups([$names])`
 Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->string()`
 Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->study($name, [$index])`
 Inherited, see “Access to the header” in Mail::Message::Head

About the body

Extends “About the body” in Mail::Message::Head::Complete.

`$obj->guessBodySize()`
 Inherited, see “About the body” in Mail::Message::Head

`$obj->guessTimeStamp()`
 Inherited, see “About the body” in Mail::Message::Head::Complete

`$obj->isMultipart()`
 Inherited, see “About the body” in Mail::Message::Head

`$obj->recvstamp()`
 Inherited, see “About the body” in Mail::Message::Head::Complete

`$obj->timestamp()`
 Inherited, see “About the body” in Mail::Message::Head::Complete

Internals

Extends “Internals” in Mail::Message::Head::Complete.

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

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

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

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

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

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

`$obj->messageIdPrefix([$prefix, [$hostname]|CODE])`
 Mail::Message::Replace::MailHeader->messageIdPrefix([\$prefix, [\$hostname]|CODE])
 Inherited, see “Internals” in Mail::Message::Head::Complete

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

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

Error handling

Extends “Error handling” in Mail::Message::Head::Complete.

`$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::Message::Replace::MailHeader->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::MailHeader->log([$level, [$strings]])`

Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`

`Mail::Message::Replace::MailHeader->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::Message::Head::Complete.

`$obj->DESTROY()`

Inherited, see “Cleanup” in Mail::Reporter

Access to the header

Simulating Mail::Header

`$obj->as_string()`

Returns the whole header as one big scalar. Calls `Mail::Message::Head::Complete::string()`.

`$obj->cleanup()`

Cleanup memory usage. Not needed here.

`$obj->combine($tag, [$with])`

I do not see any valid reason for this command, so did not implement it.

`$obj->dup()`

Duplicate the header, which is simply `clone()`.

`$obj->empty()`

Clean-out the whole hash. Better not use this (simply create another header object), although it should work.

`$obj->exists()`

Returns whether there are any fields.

`$obj->extract(ARRAY)`

Extract (and remove) header fields from the array.

`$obj->fold([$length])`

Refold all fields in the header, to `$length` or whatever `fold_length()` returns.

`$obj->fold_length([$tag], $length)`

Returns the line wrap, optionally after setting it to `$length`. The old value is returned. The `$tag` argument is ignored, because it is silly to have different lines fold in different ways. This method cannot be called statically anymore.

`$obj->header([ARRAY])`

Extract the fields from the ARRAY, if specified, and then fold the fields. Returned is an array with all fields, produced via `orderedFields()`.

`$obj->header_hashref(HASH)`

If you are using this method, you must be stupid... anyway: I do not want to support it for now: use `add()` and friends.

`$obj->mail_from(['IGNORE'|'ERROR'|'COERCE'|'KEEP'])`

What to do when a header line in the form 'From ' is encountered. Valid values are `IGNORE` – ignore and discard the header, `ERROR` – invoke an error (call `die`), `COERCE` – rename them as Mail-From and `KEEP` – keep them.

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

Refold the headers when they are added.

`$obj->read($file)`

Read the header from the `$file`.

`$obj->tags()`

Returns all the names of fields, implemented by `Mail::Message::Head::Complete::names()`.

`$obj->unfold([$tag])`

Remove the folding for all instances of `$tag`, or all fields at once.

The nasty bits

`$obj->isa($class)`

`Mail::Message::Replace::MailHeader->isa($class)`

Of course, the `isa()` class inheritance check should not see our nasty trick.

DETAILS

Extends “DETAILS” in `Mail::Message::Head::Complete`.

DIAGNOSTICS

Warning: Cannot remove field `$name` from header: not found.

You ask to remove a field which is not known in the header. Using `delete()`, `reset()`, or `set()` to do the job will not result in warnings: those methods check the existence of the field first.

Warning: Field objects have an implied name (`$name`)

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/>