

NAME

Mail::Message::Head::Partial – subset of header information of a message

INHERITANCE

```
Mail::Message::Head::Partial
  is a Mail::Message::Head::Complete
  is a Mail::Message::Head
  is a Mail::Reporter
```

SYNOPSIS

```
my $partial = $head->strip;
$partial->isa('Mail::Message::Head') # true
$partial->isDelayed                    # false
$partial->isPartial                    # true

$partial->removeFields( qr/^X-/ );
$partial->removeFieldsExcept( qw/To From/ );
$partial->removeResentGroups;
$partial->removeListGroup;
$partial->removeSpamGroups;
```

DESCRIPTION

Header information consumes a considerable amount of memory. Most of this information is only useful during a short period of time in your program, or sometimes it is not used at all. You then can decide to remove most of the header information. However, be warned that it will be lost permanently: the header (and therefore the message) gets mutilated!

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::Head::Partial->**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::Head::Partial->**new**(%options)

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

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.

`$obj->add($field | $line | <$name, $body, [$attrs]>)`

Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->addListGroup($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->cleanupOrderedFields()`

The header maintains a list of fields which are ordered in sequence of definition. It is required to maintain the header order to keep the related fields of resent groups together. The fields are also included in a hash, sorted on their name for fast access.

The references to field objects in the hash are real, those in the ordered list are weak. So when field objects are removed from the hash, their references in the ordered list are automatically undef'd.

When many fields are removed, for instance with **removeFields()** or **removeFieldsExcept()**, then it is useful to remove the list of undefs from the ordered list as well. In those cases, this method is called automatically, however you may have your own reasons to call this method.

`$obj->count($name)`

Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->delete($name)`

Inherited, see “Access to the header” in Mail::Message::Head::Complete

`$obj->get($name, [$index])`

Inherited, see “Access to the header” in Mail::Message::Head

`$obj->grepNames([@names|ARRAY|Regexprs])`

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|Regexp>, ...)`

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|Regexp>, ...)`

Remove the fields from the header which are exactly named 'STRING' (case insensitive) or match the regular expression. Do not forget to add the 'i' modifier to the regexp, because fields are case insensitive.

See also **removeField()** which is used to remove one field object from the header. The reverse specification can be made with `removeFieldsExcept()`.

example:

```
$head->removeFields('bcc', 'received');
$head->removeFields( qr/^content-/i );
```

`$obj->removeFieldsExcept(STRING|Regexp, ...)`

Remove all fields from the header which are not equivalent to one of the specified STRINGS (case-insensitive) and which are not matching one of the REGular EXPressions. Do not forget to add the 'i' modifier to the Regexp, because fields are case insensitive.

See also **removeField()** which is used to remove one field object from the header. The reverse specification can be made with `removeFields()`.

example:

```
$head->removeFieldsExcept('subject', qr/^content-/i );
$head->removeFieldsExcept( qw/subject to from sender cc/ );
```

`$obj->removeListGroup()`

Removes all header lines which are used to administer mailing lists. Which fields that are is explained in Mail::Message::Head::ListGroup. Returned is the number of removed lines.

`$obj->removeResentGroups()`

Removes all header lines which are member of a *resent group*, which are explained in Mail::Message::Head::ResentGroup. Returned is the number of removed lines.

For removing single groups (for instance because you want to keep the last), use **Mail::Message::Head::FieldGroup::delete()**.

`$obj->removeSpamGroups()`

Removes all header lines which were produced by spam detection and spam-fighting software. Which fields that are is explained in Mail::Message::Head::SpamGroup. Returned is the number of removed lines.

`$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::Head::Partial->messageIdPrefix([\$prefix, [\$hostname]]CODE])

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

`$obj->moveLocation($distance)`

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

`$obj->read($parser)`

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::Head::Partial->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::Head::Partial->log([$level, [$strings]])`

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

`$obj->logPriority($level)`

`Mail::Message::Head::Partial->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

DETAILS

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

Ordered header fields

Extends “Ordered header fields” in Mail::Message::Head::Complete.

Head class implementation

Extends “Head class implementation” in Mail::Message::Head::Complete.

Subsets of header fields

Extends “Subsets of header fields” in Mail::Message::Head::Complete.

Reducing the header size

A message header is very large in memory and quite large on disk, and therefore a good candidate for size reduction. One way to reduce the size is by simply eliminating superfluous header fields. Each field requires at least 100 bytes of run-time memory, so this may help!

Before you start playing around with **removeFields()** and **removeFieldsExcept()**, you may take a look at two large groups of fields which can be removed as sets: the resent headers and the mailinglist headers.

Resent headers describe the intermediate steps in the transmission process for the messages. After successful delivery, they are rarely useful.

When you are archiving a mailinglist, it is hardly ever useful to store the list administration lines for each message as well.

. see examples/reduce.pl in distribution

```
foreach my $msg ($folder->messages)
{
    $msg->head->removeResentGroups;
    $msg->head->removeResentList;
}
```

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/>