NAME

Mail::Message::Construct::Read − read a Mail::Message from a file handle

SYNOPSIS

```
 my $msg1 = Mail::Message−>read(\*STDIN);
 my $msg2 = Mail::Message−>read(\@lines);
```

DESCRIPTION

When complex methods are called on a `Mail::Message` object, this package is autoloaded to support the reading of messages directly from any file handle.

METHODS

Constructing a message

Mail::Message−>**read**($fh|STRING|SCALAR|ARRAY, `%options`)

Read a message from a `$fh`, STRING, SCALAR, or a reference to an ARRAY of lines. Most `%options` are passed to the **new()** of the message which is created, but a few extra are defined.

Please have a look at **build()** and **buildFromBody()** before thinking about this `read` method. Use this `read` only when you have a file-handle like STDIN to parse from, or some external source of message lines. When you already have a separate set of head and body lines, then `read` is certainly **not** your best choice.

Some people use this method in a procmail script: the message arrives at stdin, so we only have a filehandle. In this case, you are stuck with this method. The message is preceded by a line which can be used as message separator in mbox folders. See the example how to handle that one.

This method will remove `Status` and `X−Status` fields when they appear in the source, to avoid the risk that these fields accidentally interfere with your internal administration, which may have security implications.

```
 −Option              −−Default
  body_type             undef
  strip_status_fields  <true>
```

body_type => CLASS

Force a body type (any specific implementation of a Mail::Message::Body) to be used to store the message content. When the body is a multipart or nested, this will be overruled.

strip_status_fields => BOOLEAN

Remove the `Status` and `X−Status` fields from the message after reading, to lower the risk that received messages from external sources interfere with your internal administration. If you want fields not to be stripped (you would like to disable the stripping) you probably process folders yourself, which is a Bad Thing!

example:

```
 my $msg1 = Mail::Message−>read(\*STDIN);
 my $msg2 = Mail::Message−>read(\@lines, log => 'PROGRESS');
 $folder−>addMessages($msg1, $msg2);

 my $msg3 = Mail::Message−>read(<<MSG);
 Subject: hello world
 To: you@example.com
                      # warning: empty line required !!!
 Hi, greetings!
 MSG

 # promail example
 my $fromline = <STDIN>;
 my $msg      = Mail::Message−>read(\*STDIN);
```

```
    my $coerced  = $mboxfolder->addMessage($msg);
    $coerced->fromLine($fromline);
```

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