

NAME

Mail::Box::Parser::Perl – reading messages from file using Perl

INHERITANCE

```
Mail::Box::Parser::Perl
  is a Mail::Box::Parser
  is a Mail::Reporter
```

SYNOPSIS**DESCRIPTION**

The `Mail::Box::Parser::Perl` implements parsing of messages in Perl. This may be a little slower than the C based parser `Mail::Box::Parser::C`, but will also work on platforms where no C compiler is available.

Extends “DESCRIPTION” in `Mail::Box::Parser`.

METHODS

Extends “METHODS” in `Mail::Box::Parser`.

Constructors

Extends “Constructors” in `Mail::Box::Parser`.

`Mail::Box::Parser::Perl->new(%options)`

-Option	--Defined in	--Default
<code>file</code>	<code>Mail::Box::Parser</code>	<code>undef</code>
<code>filename</code>	<code>Mail::Box::Parser</code>	<code><required></code>
<code>fix_header_errors</code>		<code><false></code>
<code>log</code>	<code>Mail::Reporter</code>	<code>'WARNINGS'</code>
<code>mode</code>	<code>Mail::Box::Parser</code>	<code>'r'</code>
<code>trace</code>	<code>Mail::Reporter</code>	<code>'WARNINGS'</code>
<code>trusted</code>		<code><false></code>

`file => FILE-HANDLE`

`filename => FILENAME`

`fix_header_errors => BOOLEAN`

When header errors are detected, the parsing of the header will be stopped. Other header lines will become part of the body of the message. Set this flag to have the erroneous line added to the previous header line.

`log => LEVEL`

`mode => OPENMODE`

`trace => LEVEL`

`trusted => BOOLEAN`

Is the input from the file to be trusted, or does it require extra tests. Related to `Mail::Box::new(trusted)`.

The parser

Extends “The parser” in `Mail::Box::Parser`.

`$obj->fileChanged()`

Inherited, see “The parser” in `Mail::Box::Parser`

`$obj->filename()`

Inherited, see “The parser” in `Mail::Box::Parser`

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

If set to `true`, parsing of a header will not stop on an error, but attempt to add the erroneous this line to previous field. Without `BOOLEAN`, the current setting is returned.

example:

```

    $folder->parser->fixHeaderErrors(1);
    my $folder = $mgr->open('folder', fix_header_errors => 1);
$obj->restart()
    Inherited, see “The parser” in Mail::Box::Parser
$obj->start(%options)
    Inherited, see “The parser” in Mail::Box::Parser
$obj->stop()
    Inherited, see “The parser” in Mail::Box::Parser

```

Parsing

Extends “Parsing” in Mail::Box::Parser.

```

$obj->bodyAsFile( $fh [$chars, [$lines]] )
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->bodyAsList( [$chars, [$lines]] )
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->bodyAsString( [$chars, [$lines]] )
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->bodyDelayed( [$chars, [$lines]] )
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->filePosition( [$position] )
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->lineSeparator()
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->popSeparator()
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->pushSeparator(STRING|Regexp)
    Inherited, see “Parsing” in Mail::Box::Parser
$obj->readHeader()
$obj->readSeparator(%options)
    Inherited, see “Parsing” in Mail::Box::Parser

```

Internals

Extends “Internals” in Mail::Box::Parser.

```

$obj->closeFile()
    Inherited, see “Internals” in Mail::Box::Parser
$obj->defaultParserType( [$class] )
Mail::Box::Parser::Perl->defaultParserType( [$class] )
    Inherited, see “Internals” in Mail::Box::Parser
$obj->openFile($args)
    Inherited, see “Internals” in Mail::Box::Parser
$obj->takeFileInfo()
    Inherited, see “Internals” in Mail::Box::Parser

```

Error handling

Extends “Error handling” in Mail::Box::Parser.

```

$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::Box::Parser::Perl->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::Parser::Perl->log([\$level, [\$strings]])
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`
 Mail::Box::Parser::Perl->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::Box::Parser.

`$obj->DESTROY()`
 Inherited, see “Cleanup” in Mail::Reporter

DIAGNOSTICS

Warning: File `$filename` changed during access.

When a message parser starts working, it takes size and modification time of the file at hand. If the folder is written, it checks whether there were changes in the file made by external programs.

Calling **Mail::Box::update()** on a folder before it being closed will read these new messages. But the real source of this problem is locking: some external program (for instance the mail transfer agent, like sendmail) uses a different locking mechanism as you do and therefore violates your rights.

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.

Warning: Unexpected end of header in `$source: $line`

While parsing a message from the specified source (usually a file name), the parser found a syntax error. According to the MIME specification in the RFCs, each header line must either contain a colon, or start with a blank to indicate a folded field. Apparently, this header contains a line which starts on the first position, but not with a field name.

By default, parsing of the header will be stopped. If there are more header lines after the erroneous

line, they will be added to the body of the message. In case of `new(fix_headers)` set, the parsing of the header will be continued. The erroneous line will be added to the preceding field.

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/>