

**NAME**

Mail::Transport::Receive – receive messages

**INHERITANCE**

```
Mail::Transport::Receive
  is a Mail::Transport
  is a Mail::Reporter
```

```
Mail::Transport::Receive is extended by
  Mail::Transport::IMAP4
  Mail::Transport::POP3
```

**SYNOPSIS**

```
my $receiver = Mail::Transport::POP3->new(...);
my $message = $receiver->receive($id);
```

**DESCRIPTION**

Each object which extends Mail::Transport::Receive implement a protocol which can get messages into your application. The internals of each implementation can differ quite a lot, so have a look at each separate manual page as well.

Current message receivers:

- Mail::Transport::POP3

Implements the POP3 protocol. See also Mail::Box::POP3.

Extends “DESCRIPTION” in Mail::Transport.

**METHODS**

Extends “METHODS” in Mail::Transport.

**Constructors**

Extends “Constructors” in Mail::Transport.

Mail::Transport::Receive->**new**(%options)

Inherited, see “METHODS” in Mail::Transport

**Receiving mail**

\$obj->**receive**( [\$unique\_message\_id] )

Receive one message from the remote server. Some receivers will provide the next message automatically, other are random access and use the specified ID.

**Server connection**

Extends “Server connection” in Mail::Transport.

\$obj->**findBinary**( \$name, [@directories] )

Inherited, see “Server connection” in Mail::Transport

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

Inherited, see “Server connection” in Mail::Transport

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

Inherited, see “Server connection” in Mail::Transport

**Error handling**

Extends “Error handling” in Mail::Transport.

\$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::Transport::Receive->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::Transport::Receive->log( [\$level, [\$strings]] )

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

`$obj->logPriority($level)`

Mail::Transport::Receive->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::Transport.

`$obj->DESTROY()`

Inherited, see “Cleanup” in Mail::Reporter

## DIAGNOSTICS

Warning: Avoid program abuse: specify an absolute path for `$exec`.

Specifying explicit locations for executables of email transfer agents should only be done with absolute file names, to avoid various potential security problems.

Warning: Executable `$exec` does not exist.

The explicitly indicated mail transfer agent does not exist. The normal settings are used to find the correct location.

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-Transport distribution version 3.005, built on July 22, 2020. Website: <http://perl.overmeer.net/CPAN/>

## LICENSE

Copyrights 2001–2020 by [Mark Overmeer]. 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/>