

NAME

Mail::Transport – use Mail Transfer Agents (MTAs)

INHERITANCE

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

```
Mail::Transport is extended by
    Mail::Transport::Receive
    Mail::Transport::Send
```

SYNOPSIS

```
my $message = Mail::Message->new(...);

# Some extensions implement sending:
$message->send;
$message->send(via => 'sendmail');

my $sender = Mail::Transport::SMTP->new(...);
$sender->send($message);

# Some extensions implement receiving:
my $receiver = Mail::Transport::POP3->new(...);
$message = $receiver->receive;
```

DESCRIPTION

Objects which extend Mail::Transport implement sending and/or receiving of messages, using various protocols.

Mail::Transport::Send extends this class, and offers general functionality for send protocols, like SMTP. Mail::Transport::Receive also extends this class, and offers receive method. Some transport protocols will implement both sending and receiving.

Extends “DESCRIPTION” in Mail::Reporter.

METHODS

Extends “METHODS” in Mail::Reporter.

Constructors

Extends “Constructors” in Mail::Reporter.

```
Mail::Transport->new(%options)
    -Option      --Defined in      --Default
    executable
    hostname
    interval
    log           Mail::Reporter    'WARNINGS'
    password
    port
    proxy
    retry
    timeout
    trace         Mail::Reporter    'WARNINGS'
    username
    via           'sendmail'
```

executable => FILENAME

If you specify an executable, the module does not need to search the system directories to figure-out where the client lives. Using this decreases the flexible usage of your program: moving your program to other systems may involve changing the path to the executable, which otherwise would

work auto-detect and unmodified.

hostname => HOSTNAME|ARRAY

The host on which the server runs. Some protocols accept an ARRAY of alternatives for this option.

interval => SECONDS

The time between tries to contact the remote server for sending or receiving a message in SECONDS. This number must be larger than 0.

log => LEVEL

password => STRING

Some protocols require a password to be given, usually in combination with a password.

port => INTEGER

The port number behind which the service is hiding on the remote server.

proxy => PATH

The name of the proxy software (the protocol handler). This must be the name (preferable the absolute path) of your mail delivery software.

retry => NUMBER|undef

The number of retries before the sending will fail. If undef, the number of retries is unlimited.

timeout => SECONDS

SECONDS till time-out while establishing the connection to a remote server.

trace => LEVEL

username => STRING

Some protocols require a user to login.

via => CLASS|NAME

Which CLASS (extending Mail::Transport) will transport the data. Some predefined NAMES avoid long class names: mail and mailx are handled by the Mail::Transport::Mailx module, sendmail and postfix belong to Mail::Transport::Sendmail, and smtp is implemented in Mail::Transport::SMTP.

The pop or pop3 protocol implementation can be found distribution Mail::Transport::POP3. For imap or imap4, install Mail::Transport::IMAP4.

Server connection

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

Look for a binary with the specified \$name in the directories which are defined to be safe. The list of standard directories is followed by the optional @directories. The full pathname is returned.

You may specify new(proxy), which specifies the absolute name of the binary to be used.

`$obj->remoteHost()`

Returns the hostname, port number, username and password to be used to establish the connection to the server for sending or receiving mail.

`$obj->retry()`

Returns the retry interval, retry count, and timeout for the connection.

Error handling

Extends “Error handling” in Mail::Reporter.

`$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->**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->**log**([\$level, [\$strings]])

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

\$obj->**logPriority**(\$level)

Mail::Transport->**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::Reporter.

\$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/>