

**NAME**

Mail::Transport::SMTP – transmit messages without external program

**INHERITANCE**

```
Mail::Transport::SMTP
  is a Mail::Transport::Send
  is a Mail::Transport
  is a Mail::Reporter
```

**SYNOPSIS**

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

$message->send(via => 'smtp');
```

**DESCRIPTION**

This module implements transport of Mail::Message objects by negotiating to the destination host directly by using the SMTP protocol, without help of sendmail, mail, or other programs on the local host.

**warning:** you may need to install Net::SMTPS, to get TLS support.

Extends “DESCRIPTION” in Mail::Transport::Send.

**METHODS**

Extends “METHODS” in Mail::Transport::Send.

**Constructors**

Extends “Constructors” in Mail::Transport::Send.

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

-Option	--Defined in	--Default
esmtplib_options		{ }
executable	Mail::Transport	undef
from		undef
helo		<from Net::Config>
hostname	Mail::Transport	<from Net::Config>
interval	Mail::Transport	30
log	Mail::Reporter	'WARNINGS'
password		undef
port	Mail::Transport	25
proxy	Mail::Transport	<from Net::Config>
retry	Mail::Transport	<false>
smtp_debug		<false>
timeout		120
trace	Mail::Reporter	'WARNINGS'
username		undef
via	Mail::Transport	'smtp'

esmtplib\_options => HASH

[2.116] ESMTP options to pass to Net::SMTP. See the Net::SMTP documentation for full details.

Options can also be passed at send time. For example: { XVERP => 1 }

executable => FILENAME

from => ADDRESS

Allows a default sender address to be specified globally. See **trySend()** for full details.

helo => HOST

The fully qualified name of the sender’s host (your system) which is used for the greeting message to the receiver. If not specified, Net::Config or else Net::Domain are questioned to find it. When even these do not supply a valid name, the name of the domain in the From line of the message is

assumed.

hostname => HOSTNAME|ARRAY

interval => SECONDS

log => LEVEL

password => STRING

The password to be used with the new(username) to log in to the remote server.

port => INTEGER

proxy => PATH

retry => NUMBER|undef

smtp\_debug => BOOLEAN

Simulate transmission: the SMTP protocol output will be sent to your screen.

timeout => SECONDS

The number of seconds to wait for a valid response from the server before failing.

trace => LEVEL

username => STRING

Use SASL authentication to contact the remote SMTP server (RFC2554). This username in combination with new(password) is passed as arguments to Net::SMTP method auth. Other forms of authentication are not supported by Net::SMTP. The username can also be specified as an Authn::SASL object.

via => CLASS|NAME

### **Sending mail**

Extends “Sending mail” in Mail::Transport::Send.

\$obj->**destinations**( \$message, [\$address|ARRAY] )

Inherited, see “Sending mail” in Mail::Transport::Send

\$obj->**putContent**(\$message, \$fh, %options)

Inherited, see “Sending mail” in Mail::Transport::Send

\$obj->**send**(\$message, %options)

Inherited, see “Sending mail” in Mail::Transport::Send

\$obj->**trySend**(\$message, %options)

Try to send the \$message once. This may fail, in which case this method will return false. In list context, the reason for failure can be caught: in list context trySend will return a list of six values:

```
(success, rc, rc-text, error location, quit success, accept)
```

Success and quit success are booleans. The error code and -text are protocol specific codes and texts. The location tells where the problem occurred.

[3.003] the 'accept' returns the message of the **dataend()** instruction. Some servers may provide useful information in there, like an internal message registration id. For example, postfix may return “2.0.0 Ok: queued as 303EA380EE”. You can only use this parameter when running local delivery (which is a smart choice anyway)

-Option	--Default
esmtplib_options	{ }
from	< >
to	[ ]

esmtplib\_options => HASH

Additional or overridden EMSTP options. See new(esmtplib\_options)

from => ADDRESS

Your own identification. This may be fake. If not specified, it is taken from **Mail::Message::sender()**, which means the content of the Sender field of the message or the first address of the From field. This defaults to “<>”, which represents “no address”.

to => ADDRESS|[ADDRESSES]

Alternative destinations. If not specified, the To, Cc and Bcc fields of the header are used. An address is a string or a Mail::Address object.

### Server connection

Extends “Server connection” in Mail::Transport::Send.

`$obj->contactAnyServer()`

Creates the connection to the SMTP server. When more than one hostname was specified, the first which accepts a connection is taken. An IO::Socket::INET object is returned.

`$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

`$obj->tryConnectTo($host, %options)`

Try to establish a connection to deliver SMTP to the specified \$host. The %options are passed to the new method of Net::SMTP.

### Error handling

Extends “Error handling” in Mail::Transport::Send.

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

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

`$obj->logPriority($level)`

`Mail::Transport::SMTP->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::Send.

`$obj->DESTROY()`

Inherited, see “Cleanup” in Mail::Reporter

**DIAGNOSTICS**

Warning: Message has no destination

It was not possible to figure-out where the message is intended to go to.

Notice: No addresses found to send the message to, no connection made

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: Resent group does not specify a destination

The message which is sent is the result of a bounce (for instance created with **Mail::Message::bounce()**), and therefore starts with a `Received` header field. With the bounce, the new destination(s) of the message are given, which should be included as `Resent-To`, `Resent-Cc`, and `Resent-Bcc`.

The `To`, `Cc`, and `Bcc` header information is only used if no `Received` was found. That seems to be the best explanation of the RFC.

As alternative, you may also specify the `to` option to some of the senders (for instance `Mail::Transport::SMTP::send(to)` to overrule any information found in the message itself about the destination.

**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/>