

## NAME

Mail::Transport::Send – send a message

## INHERITANCE

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

```
Mail::Transport::Send is extended by
Mail::Transport::Exim
Mail::Transport::Mailx
Mail::Transport::Qmail
Mail::Transport::SMTP
Mail::Transport::Sendmail
```

## 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);
```

## DESCRIPTION

Send a message to the destinations as specified in the header. The `Mail::Transport::Send` module is capable of autodetecting which of the following modules work on your system; you may simply call `send` without `via` options to get a message transported.

- `Mail::Transport::Sendmail`  
Use `sendmail` to process and deliver the mail. This requires the `sendmail` program to be installed on your system. Whether this is an original `sendmail`, or a replacement from Postfix does matter.
- `Mail::Transport::Exim`  
Use `exim` to distribute the message.
- `Mail::Transport::Qmail`  
Use `qmail-inject` to distribute the message.
- `Mail::Transport::SMTP`  
In this case, Perl is handling mail transport on its own. This is less desired but more portable than sending with `sendmail` or `qmail`. The advantage is that this sender is environment independent, and easier to configure. However, there is no daemon involved which means that your program will wait until the message is delivered, and the message is lost when your program is interrupted during delivery (which may take hours to complete).
- `Mail::Transport::Mailx`  
Use the external `mail`, `mailx`, or `Mail` programs to send the message. Usually, the result is poor, because some versions of these programs do not support MIME headers. Besides, these programs are known to have exploitable security breaches.

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

## METHODS

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

**Constructors**

Extends “Constructors” in Mail::Transport.

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

-Option	--Defined in	--Default
executable	Mail::Transport	undef
hostname	Mail::Transport	'localhost'
interval	Mail::Transport	30
log	Mail::Reporter	'WARNINGS'
password	Mail::Transport	undef
port	Mail::Transport	undef
proxy	Mail::Transport	undef
retry	Mail::Transport	<false>
timeout	Mail::Transport	120
trace	Mail::Reporter	'WARNINGS'
username	Mail::Transport	undef
via	Mail::Transport	'sendmail'

executable => FILENAME

hostname => HOSTNAME|ARRAY

interval => SECONDS

log => LEVEL

password => STRING

port => INTEGER

proxy => PATH

retry => NUMBER|undef

timeout => SECONDS

trace => LEVEL

username => STRING

via => CLASS|NAME

**Sending mail**

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

Determine the destination for this message. If a valid \$address is defined, this is used to overrule the addresses within the message. If the \$address is undef it is ignored. It may also be an ARRAY of addresses.

If no \$address is specified, the message is scanned for resent groups (see **Mail::Message::Head::Complete::resentGroups()**). The addresses found in the first (is latest added) group are used. If no resent groups are found, the normal To, Cc, and Bcc lines are taken.

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

Print the content of the \$message to the \$fh.

-Option	--Default
body_only	<false>
undisclosed	<false>

body\_only => BOOLEAN

Print only the body of the message, not the whole.

undisclosed => BOOLEAN

Do not print the Bcc and Resent-Bcc lines. Default false, which means that they are not printed.

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

Transmit the \$message, which may be anything what can be coerced into a Mail::Message, so including Mail::Internet and MIME::Entity messages. It returns true when the transmission was successfully completed.

```

-Option  --Default
interval new(interval)
retry    new(retry)
to       undef

```

interval => SECONDS

retry => INTEGER

to => STRING

Overrules the destination(s) of the message, which is by default taken from the (Resent-)To, (Resent-)Cc, and (Resent-)Bcc.

`$obj->trySend($message, %options)`

Try to send the message. This will return true if successful, and false in case some problems were detected. The \$? contains the exit status of the command which was started.

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

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

`$obj->logPriority($level)`

`Mail::Transport::Send->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: Message has no destination

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

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.

Error: Transporters of type `$class` cannot send.

The Mail::Transport object of the specified type can not send messages, but only receive message.

## 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/>