

**NAME**

Mail::Transport::Mailx – transmit messages using external mailx program

**INHERITANCE**

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

**SYNOPSIS**

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

**DESCRIPTION**

Implements mail transport using the external programs 'mailx', 'Mail', or 'mail'. When instantiated, the mailer will look for any of these binaries in specific system directories, and the first program found is taken.

**WARNING: There are many security issues with mail and mailx. DO NOT USE these commands to send messages which contains data derived from any external source!!!**

Under Linux, freebsd, and bsdos the mail, Mail, and mailx names are just links to the same binary. The implementation is very primitive, pre-MIME standard, what may cause many headers to be lost. For these platforms (and probably for other platforms as well), you can better not use this transport mechanism.

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

**METHODS**

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

**Constructors**

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

Mail::Transport::Mailx->**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>
style		<autodetect>
timeout	Mail::Transport	120
trace	Mail::Reporter	'WARNINGS'
username	Mail::Transport	undef
via	Mail::Transport	'mailx'

executable => FILENAME

hostname => HOSTNAME|ARRAY

interval => SECONDS

log => LEVEL

password => STRING

port => INTEGER

proxy => PATH

retry => NUMBER|undef

style => 'BSD'|'RFC822'

There are two version of the mail program. The newest accepts RFC822 messages, and automagically collect information about where the message is to be send to. The BSD style mail

command predates MIME, and expects lines which start with a '~' (tilde) to specify destinations and such. This field is autodetect, however on some platforms both versions of mail can live (like various Linux distributions).

```
timeout => SECONDS
trace => LEVEL
username => STRING
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)
```

### **Server connection**

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

```
$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::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, $callback] ] )
Mail::Transport::Mailx->defaultTrace( [$level][[$loglevel, $tracelevel][[$level, $callback] ] )
    Inherited, see “Error handling” in Mail::Reporter

$obj->errors()
    Inherited, see “Error handling” in Mail::Reporter

$obj->log( [$level, [$strings]] )
Mail::Transport::Mailx->log( [$level, [$strings]] )
    Inherited, see “Error handling” in Mail::Reporter

$obj->logPriority($level)
Mail::Transport::Mailx->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.

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: Sending via mailx mailer `$program` failed: `$( $?)`

Mailx (in some shape: there are many different implementations) did start accepting messages, but did not succeed sending it.

**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/>