



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'Mail::Util.3pm'

\$ man Mail::Util.3pm

Mail::Util(3pm) User Contributed Perl Documentation Mail::Util(3pm)

NAME

Mail::Util - mail utility functions

INHERITANCE

Mail::Util

is a Exporter

SYNOPSIS

use Mail::Util qw(...);

DESCRIPTION

This package provides several mail related utility functions. Any function required must by explicitly listed on the use line to be exported into the calling package.

FUNCTIONS

mailaddress([\$address])

Return a guess at the current users mail address. The user can force the return value by setting the MAILADDRESS environment variable. [2.10] You may set the \$address via the parameter.

WARNING: When not supplied via the environment variable, <mailaddress> looks at

various configuration files and other environmental data. Although this seems to be smart behavior, this is not predictable enough (IMHO) to be used. Please set the MAILADDRESS explicitly, and do not trust on the "automatic detection", even when that produces a correct address (on the moment)

example:

```
# in your main script
$ENV{MAILADDRESS} = 'me@example.com';

# everywhere else
use Mail::Util 'mailaddress';
print mailaddress;

# since v2.10
mailaddress "me@example.com";
```

maildomain()

Attempt to determine the current user mail domain string via the following methods

- ? Look for the MAILDOMAIN environment variable, which can be set from outside the program. This is by far the best way to configure the domain.
- ? Look for a sendmail.cf file and extract DH parameter
- ? Look for a smail config file and use the first host defined in hostname(s)
- ? Try an SMTP connect (if Net::SMTP exists) first to mailhost then localhost
- ? Use value from Net::Domain::domainname (if Net::Domain exists)

WARNING: On modern machines, there is only one good way to provide information to this method: the first; always explicitly configure the MAILDOMAIN.

example:

```
# in your main script
```

```
$ENV{MAILDOMAIN} = 'example.com';
```

```
# everywhere else
```

```
use Mail::Util 'maildomain';
```

```
print maildomain;
```

read_mbox(\$file)

Read \$file, a binmail mailbox file, and return a list of references. Each reference is a reference to an array containing one message.

WARNING: This method does not quote lines which accidentally also start with the message separator "From", so this implementation can be considered broken. See

Mail::Box::Mbox

SEE ALSO

This module is part of the MailTools distribution, <http://perl.overmeer.net/mailtools/>.

AUTHORS

The MailTools bundle was developed by Graham Barr. Later, Mark Overmeer took over maintenance without commitment to further development.

Mail::Cap by Gisle Aas <aas@oslonett.no>. Mail::Field::AddrList by Peter Orbaek <poe@cit.dk>. Mail::Mailer and Mail::Send by Tim Bunce <Tim.Bunce@ig.co.uk>. For other contributors see ChangeLog.

LICENSE

Copyrights 1995-2000 Graham Barr <gbarr@pobox.com> and 2001-2017 Mark Overmeer <perl@overmeer.net>.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself. See <http://www.perl.com/perl/misc/Artistic.html>

perl v5.28.1

2019-07-25

Mail::Util(3pm)