NAME

Mail::Internet – manipulate email messages

SYNOPSIS

```
use Mail::Internet;
my $msg = Mail::Internet->new(\*STDIN);
```

DESCRIPTION

This package implements reading, creating, manipulating, and writing email messages. Sometimes, the implementation tries to be too smart, but in the general case it works as expected.

If you start writing a **new application**, you should use the Mail::Box distribution, which has more features and handles messages much better according to the RFCs. See <http://perl.overmeer.net/mailbox/>. You may also chose MIME::Entity, to get at least some multipart support in your application.

METHODS

Constructors

`$obj->`**dup**()

Duplicate the message as a whole. Both header and body will be deep-copied: a new Mail::Internet object is returned.

`$obj->`**extract**(\@lines)

Extract header and body from an ARRAY of message lines. Requires an object already created with **new()**, which contents will get overwritten.

`$obj->`**new**([$arg], [%options])

Mail::Internet**->new**([$arg], [%options])

`$arg` is optional and may be either a file descriptor (reference to a GLOB) or a reference to an array. If given the new object will be initialized with headers and body either from the array of read from the file descriptor.

The **Mail::Header::new()** `%options` `Modify`, `MailFrom` and `FoldLength` may also be given.

```
-Option--Default
 Body    []
 Header  undef
```

Body => ARRAY-of-LINES

The value of this option should be a reference to an array which contains the lines for the body of the message. Each line should be terminated with `\n` (LF). If Body is given then `Mail::Internet` will not attempt to read the body from `$arg` (even if it is specified).

Header => Mail::Header

The value of this option should be a Mail::Header object. If given then `Mail::Internet` will not attempt to read a mail header from `$arg`, if it was specified.

`$obj->`**read**($fh)

Read a message from the `$fh` into an already existing message object. Better use **new()** with the `$fh` as first argument.

Accessors

`$obj->`**body**([$body])

Returns the body of the message. This is a reference to an array. Each entry in the array represents a single line in the message.

If *`$body`* is given, it can be a reference to an array or an array, then the body will be replaced. If a reference is passed, it is used directly and not copied, so any subsequent changes to the array will change the contents of the body.

`$obj->`**head**()

Returns the `Mail::Header` object which holds the headers for the current message

Processing the message as a whole

`$obj->`**as_mbox_string**([$already_escaped])
: Returns the message as a string in mbox format. `$already_escaped`, if given and true, indicates that **escape_from()** has already been called on this object.

`$obj->`**as_string**()
: Returns the message as a single string.

`$obj->`**print**([$fh])
: Print the header, body or whole message to file descriptor *`$fh`*. *`$fd`* should be a reference to a GLOB. If *`$fh`* is not given the output will be sent to STDOUT.

  example:

```
$mail->print( \*STDOUT );  # Print message to STDOUT
```

`$obj->`**print_body**([$fh])
: Print only the body to the `$fh` (default STDOUT).

`$obj->`**print_header**([$fh])
: Print only the header to the `$fh` (default STDOUT).

Processing the header

Most of these methods are simply wrappers around methods provided by Mail::Header.

`$obj->`**add**(PAIRS)
: The PAIRS are field-name and field-content. For each PAIR, **Mail::Header::add()** is called. All fields are added after existing fields. The last addition is returned.

`$obj->`**combine**(`$tag`, [$with])
: See **Mail::Header::combine()**.

`$obj->`**delete**(`$tag`, [$tags])
: Delete all fields with the name `$tag`. **Mail::Header::delete()** is doing the work.

`$obj->`**fold**([$length])
: See **Mail::Header::fold()**.

`$obj->`**fold_length**([$tag], [$length])
: See **Mail::Header::fold_length()**.

`$obj->`**get**(`$tag`, [$tags])
: In LIST context, all fields with the name `$tag` are returned. In SCALAR context, only the first field which matches the earliest `$tag` is returned. **Mail::Header::get()** is called to collect the data.

`$obj->`**header**(\@lines)
: See **Mail::Header::header()**.

`$obj->`**replace**(PAIRS)
: The PAIRS are field-name and field-content. For each PAIR, **Mail::Header::replace()** is called with index 0. If a `$field` is already in the header, it will be removed first. Do not specified the same field-name twice.

Processing the body

`$obj->`**remove_sig**([$nlines])
: Attempts to remove a user's signature from the body of a message. It does this by looking for a line equal to `'-- '` within the last `$nlines` of the message. If found then that line and all lines after it will be removed. If `$nlines` is not given a default value of 10 will be used. This would be of most use in auto-reply scripts.

`$obj->`**sign**(%options)
: Add your signature to the body. **remove_sig()** will strip existing signatures first.

```
 -Option   --Default
  File       undef
  Signature  []
```

File => FILEHANDLE
Take from the FILEHANDLE all lines starting from the first `--`.

Signature => STRING|ARRAY-of-LINES

`$obj->`**tidy_body**()
Removes all leading and trailing lines from the body that only contain white spaces.

High-level functionality

`$obj->`**escape_from**()
It can cause problems with some applications if a message contains a line starting with `` `From ' ``, in particular when attempting to split a folder. This method inserts a leading `>` on any line that matches the regular expression `/^*From/`

`$obj->`**nntppost**([%options])
Post an article via NNTP. Requires Net::NNTP to be installed.

```
 -Option--Default
  Debug   <false>
  Host    <required>
  Port    119
```

Debug => BOOLEAN
Debug value to pass to Net::NNTP, see Net::NNTP

Host => HOSTNAME|Net::NNTP object
Name of NNTP server to connect to, or a Net::NNTP object to use.

Port => INTEGER
Port number to connect to on remote host

`$obj->`**reply**(%options)
Create a new object with header initialised for a reply to the current object. And the body will be a copy of the current message indented.

The `.mailhdr` file in your home directory (if exists) will be read first, to provide defaults.

```
 -Option  --Default
  Exclude   []
  Indent    '>'
  Keep      []
  ReplyAll  false
```

Exclude => ARRAY-of-FIELDS
Remove the listed FIELDS from the produced message.

Indent => STRING
Use as indentation string. The string may contain `%%` to get a single `%`, `%f` to get the first from name, `%F` is the first character of `%f`, `%l` is the last name, `%L` its first character, `%n` the whole from string, and `%I` the first character of each of the names in the from string.

Keep => ARRAY-of-FIELDS
Copy the listed FIELDS from the original message.

ReplyAll => BOOLEAN
Automatically include all To and Cc addresses of the original mail, excluding those mentioned in the Bcc list.

`$obj->`**send**([$type, [$args...]])
Send a Mail::Internet message using Mail::Mailer. `$type` and `$args` are passed on to **Mail::Mailer::new()**.

`$obj->`**smtpsend**([%options])

Send a Mail::Internet message using direct SMTP to the given ADDRESSES, each can be either a string or a reference to a list of email addresses. If none of `To`, <Cc> or `Bcc` are given then the addresses are extracted from the message being sent.

The return value will be a list of email addresses that the message was sent to. If the message was not sent the list will be empty.

Requires Net::SMTP and Net::Domain to be installed.

```
 -Option  --Default
  Bcc       undef
  Cc        undef
  Debug     <false>
  Hello     localhost.localdomain
  Host      $ENV{SMTPHOSTS}
  MailFrom  Mail::Util::mailaddress()
  Port      25
  To        undef
```

Bcc => ADDRESSES
Cc => ADDRESSES
Debug => BOOLEAN
Debug value to pass to Net::SMTP, see <Net::SMTP>

Hello => STRING
Send a HELO (or EHLO) command to the server with the given name.

Host => HOSTNAME
Name of the SMTP server to connect to, or a Net::SMTP object to use

If `Host` is not given then the SMTP host is found by attempting connections first to hosts specified in `$ENV{SMTPHOSTS}`, a colon separated list, then `mailhost` and `localhost`.

MailFrom => ADDRESS
The e−mail address which is used as sender. By default, **Mail::Util::mailaddress()** provides the address of the sender.

Port => INTEGER
Port number to connect to on remote host

To => ADDRESSES

`$obj->`**unescape_from**(())

Remove the escaping added by **escape_from()**.

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