

**NAME**

Mail::Message::Construct::Text – capture a Mail::Message as text

**SYNOPSIS**

```

my $text = $msg->string;
my $text = "$msg";    # via overload

my @text = $msg->lines;
my @text = @$lines;   # via overload

my $fh    = $msg->file;
my $line  = <$fh>;

$msg->printStructure;
```

**DESCRIPTION**

Complex functionality on Mail::Message objects is implemented in different files which are autoloaded. This file implements the functionality related to creating message replies.

**METHODS****The whole message as text****\$obj->file()**

Returns the message as file-handle.

**\$obj->lines()**

Returns the whole message as set of lines. In LIST context, copies of the lines are returned. In SCALAR context, a reference to an array of lines is returned.

**\$obj->printStructure( [\$fh|undef],[\$indent] )**

Print the structure of a message to the specified \$fh or the selected filehandle. When explicitly undef is specified as handle, then the output will be returned as string.

The message's subject and the types of all composing parts are displayed.

\$indent specifies the initial indentation string: it is added in front of each line. The \$indent must contain at least one white-space.

example:

```

my $msg = ...;
$msg->printStructure(\*OUTPUT);
```

```

$msg->printStructure;
```

```

my $struct = $msg->printStructure(undef);
```

# Possible output for one message:

```

multipart/mixed: forwarded message from Pietje Puk (1550 bytes)
  text/plain (164 bytes)
  message/rfc822 (1043 bytes)
    multipart/alternative: A multipart alternative (942 bytes)
      text/plain (148 bytes, deleted)
      text/html (358 bytes)
```

**\$obj->string()**

Returns the whole message as string.

**Flags****SEE ALSO**

This module is part of Mail-Message distribution version 3.012, built on February 11, 2022. Website: <http://perl.overmeer.net/CPAN/>

## **LICENSE**

Copyrights 2001–2022 by [Mark Overmeer <markov@cpan.org>]. 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/>