

NAME

Mail::Box::Search::Grep – select messages within a mail box like grep does

INHERITANCE

```
Mail::Box::Search::Grep
  is a Mail::Box::Search
  is a Mail::Reporter
```

SYNOPSIS

```
use Mail::Box::Manager;
my $mgr      = Mail::Box::Manager->new;
my $folder   = $mgr->open('Inbox');

my $filter   = Mail::Box::Search::Grep->new
  ( label => 'selected'
    , in  => 'BODY', match => qr/abc?d*e/
  );

my @msgs     = $filter->search($folder);

my $filter   = Mail::Box::Search::Grep->new
  ( field => 'To'
    , match => $my_email
  );

if($filter->search($message)) {...}
```

DESCRIPTION

Try to find some text strings in the header and footer of messages. Various ways to limit the search to certain header fields, the whole header, only the body, the whole message, but even binary multiparts, are provided for.

The name *grep* is derived from the UNIX tool *grep*, which means: “Get Regular Expression and Print”. Although you can search using regular expressions (the Perl way of them), you do not have to print those as result.

Extends “DESCRIPTION” in Mail::Box::Search.

METHODS

Extends “METHODS” in Mail::Box::Search.

Constructors

Extends “Constructors” in Mail::Box::Search.

Mail::Box::Search::Grep->**new**(%options)

Create a UNIX-grep like search filter.

Option	--Defined in	--Default
binaries	Mail::Box::Search	<false>
decode	Mail::Box::Search	<true>
delayed	Mail::Box::Search	<true>
deleted	Mail::Box::Search	<false>
deliver		undef
field		undef
in	Mail::Box::Search	<\$field ? 'HEAD' : C<'BODY'>>
label	Mail::Box::Search	undef
limit	Mail::Box::Search	0
log	Mail::Reporter	'WARNINGS'
logical	Mail::Box::Search	'REPLACE'
match		<required>

```

multiparts Mail::Box::Search <true>
trace      Mail::Reporter    'WARNINGS'

```

binaries => BOOLEAN

decode => BOOLEAN

delayed => BOOLEAN

deleted => BOOLEAN

deliver => undef|CODE|'DELETE'|LABEL|'PRINT'|REF-ARRAY

Store the details about where the match was found. The search may take much longer when this feature is enabled.

When an ARRAY is specified it will contain a list of references to hashes. Each hash contains the information of one match. A match in a header line will result in a line with fields `message`, `part`, and `field`, where the field is a `Mail::Message::Field` object. When the match is in the body the hash will contain a `message`, `part`, `linenr`, and `line`.

In case of a CODE reference, that routine is called for each match. The first argument is this search object and the second a reference to same hash as would be stored in the array.

The PRINT will call **printMatchedHead()** or **printMatchedBody()** when any matching header resp body line was found. The output is minimized by not reprinting the message info on multiple matches in the same message.

DELETE will flag the message to be deleted in case of a match. When a multipart's part is matched, the whole message will be flagged for deletion.

field => undef|STRING|REGEX|CODE

Not valid in combination with `in` set to BODY. The STRING is one full field name (case-insensitive). Use a REGEX to select more than one header line to be scanned. CODE is a routine which is called for each field in the header. The CODE is called with the header as first, and the field as second argument. If the CODE returns true, the message is selected.

in => 'HEAD'|'BODY'|'MESSAGE'

label => STRING

limit => NUMBER

log => LEVEL

logical => 'REPLACE'|'AND'|'OR'|'NOT'|'AND NOT'|'OR NOT'

match => STRING|REGEX|CODE

The pattern to be search for can be a REGular EXpression, or a STRING. In both cases, the match succeeds if it is found anywhere within the selected fields.

With a CODE reference, that function will be called each field or body-line. When the result is true, the details are delivered. The call formats are

```

$code->($head, $field);          # for HEAD searches
$code->($body, $linenr, $line);  # for BODY searches

```

The `$head` resp `$body` are one message's head resp. body object. The `$field` is a header line which matches. The `$line` and `$linenr` tell the matching line in the body.

Be warned that when you search in MESSAGE the code must accept both formats.

multiparts => BOOLEAN

trace => LEVEL

Searching

Extends "Searching" in Mail::Box::Search.

`$obj->inBody($part, $body)`

Inherited, see "Searching" in Mail::Box::Search

`$obj->inHead($part, $head)`
 Inherited, see “Searching” in Mail::Box::Search

`$obj->search($folder|$thread|$message|ARRAY)`
 Inherited, see “Searching” in Mail::Box::Search

`$obj->searchPart($part)`
 Inherited, see “Searching” in Mail::Box::Search

The Results

Extends “The Results” in Mail::Box::Search.

`$obj->printMatch([$fh], $match)`
`$obj->printMatchedBody($fh, $match)`
`$obj->printMatchedHead($fh, $match)`

Error handling

Extends “Error handling” in Mail::Box::Search.

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

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

`$obj->log([$level, [$strings]])`
`Mail::Box::Search::Grep->log([$level, [$strings]])`
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`
`Mail::Box::Search::Grep->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::Box::Search.

`$obj->DESTROY()`
 Inherited, see “Cleanup” in Mail::Reporter

DIAGNOSTICS

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.

SEE ALSO

This module is part of Mail-Box distribution version 3.009, built on August 18, 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/>