NAME

Mail::Box::Search − select messages within a mail box

INHERITANCE

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

Mail::Box::Search is extended by
  Mail::Box::Search::Grep
  Mail::Box::Search::SpamAssassin
  Mail::Server::IMAP4::Search
```

SYNOPSIS

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

my $filter = Mail::Box::Search::[something]−>new;
my @msgs   = $filter−>search($folder, ...);
if($filter−>search($message)) {...}
```

DESCRIPTION

This `Mail::Box::Search` class is the base class for various message scan algorithms. The selected messages can be labeled. Boolean operations on messages are supported.

Currently implemented searches:

Mail::Box::Search::Grep
Match header or body against a regular expression in a UNIX `grep` like fashion.

Mail::Box::Search::SpamAssassin
Try to detect spam, using Mail::SpamAssassin.

Mail::Box::Search::IMAP
Search an IMAP folder for special interface IMAP folders provide for it. UNDER CONSTRUCTION till Mail::Transport::IMAP4 is complete.

Extends "DESCRIPTION" in Mail::Reporter.

METHODS

Extends "METHODS" in Mail::Reporter.

Constructors

Extends "Constructors" in Mail::Reporter.

Mail::Box::Search−>**new**(%options)
Create a filter.

```
−Option    −−Defined in     −−Default
 binaries                     <false>
 decode                       <true>
 delayed                      <true>
 deleted                      <false>
 deliver                      undef
 in                           'BODY'
 label                        undef
 limit                        0
 log         Mail::Reporter   'WARNINGS'
 logical                      'REPLACE'
 multiparts                   <true>
 trace       Mail::Reporter   'WARNINGS'
```

binaries => BOOLEAN
Whether to include binary bodies in the search.

decode => BOOLEAN
Decode the messages before the search takes place. Even plain text messages can be encoded, for instance as `quoted−printable`, which may disturb the results. However, decoding will slowdown the search.

delayed => BOOLEAN
Include the delayed messages (which will be parsed) in the search. If you set this to false, you may find fewer hits.

deleted => BOOLEAN
In most cases, you will not be interested in results which are found in messages flagged to be deleted. However, with this option you can specify you want them to be searched too.

deliver => undef|CODE|’DELETE’
The exact functionality of this parameter differs per search method, so read the applicable manpage. In any case `undef` means that details are not collected for this search, which is the fastest search.

`DELETE` will flag the message to be flagged for deletion. You may also specify your own CODE reference. With an reference to an array, the information about the matches is collected as a list of hashes, one hash per match.

in => ’HEAD’|’BODY’|’MESSAGE’
Where to look for the match.

label => STRING
Mark all selected messages with the specified STRING. If this field is not specified, the message will not get a label; **search()** also returns a list of selected messages.

limit => NUMBER
Limit the search to the specified NUMBER of messages. When the NUMBER is positive, the search starts at the first message in the folder or thread. A negative NUMBER starts at the end of the folder. If the limit is set to zero, there is no limit.

log => LEVEL
logical => ’REPLACE’|’AND’|’OR’|’NOT’|’AND NOT’|’OR NOT’
Only applicable in combination with a `label`. How to handle the existing labels. In case of `REPLACE`, messages which already are carrying the label are stripped from their selection (unless they match again). With `AND`, the message must be selected by this search and already carry the label, otherwise the label will not be set. Specify `OR` to have newly selected messages added to the set of already selected messages.

`NOT` is true for messages which do not fulfil the search. The details output will still contain the places where the match was found, however those messages will complementary set of messages will be labeled and returned.

multiparts => BOOLEAN
Are multiparts to be included in the search results? Some MUA have problems handling details received from the search. When this flag is turned off, the body of multiparts will be ignored. The parts search will include the preamble and epilogue.

trace => LEVEL

Searching

`$obj`−>**inBody**($part, `$body`)
Tests whether body contains the requesting information. See the specific search module for its parameters.

`$obj`−>**inHead**($part, `$head`)

Tests whether header contains the requesting information. See the specific search module for its parameters.

`$obj`−>**search**($folder|$thread|$message|ARRAY)

Check which messages from the `$folder` (Mail::Box) match the search parameters. The matched messages are returned as list. You can also specify a `$thread` (a Mail::Box::Thread::Node), one single `$message` (a Mail::Message), or an ARRAY of messages.

Sometimes we know how only one match is needed. In this case, this searching will stop at the first match. For instance, when `limit` is `−1` or `1`, or when the search in done in scalar context.

example:

```
 my $grep = Mail::Box::Search::Grep−>new
  ( match   => 'My Name Is Nobody'
  , deliver => 'PRINT'
  );

 $grep−>search($folder);

 my $message = $folder−>message(3);
 $grep−>search($message);

 my $thread  = $message−>threadStart;
 $grep−>search($thread);
```

`$obj`−>**searchPart**($part)

Search this message `$part` for matches.

The Results

`$obj`−>**printMatch**([$fh], HASH)

Print the information about the match (see new(deliver)) in some understandable way. If no file handle is specified, the output will go to the selected filehandle (see `perldoc −f select`).

Error handling

Extends ‘‘Error handling’’ in Mail::Reporter.

`$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−>**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, [$strings]])
Mail::Box::Search−>**log**([$level, [$strings]])

Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj`−>**logPriority**($level)
Mail::Box::Search−>**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::Reporter.

`$obj->`**DESTROY**()
Inherited, see ‘‘Cleanup’’ in Mail::Reporter

DIAGNOSTICS

Error: Cannot search in body.
Th search object does not implement **inBody()**, and can therefore not search a message body.

Error: Cannot search in header.
Th search object does not implement **inHead()**, and can therefore not search a message header.

Error: Don’t know how to deliver via results in `$way`.
The search results cannot be delivered in the specific way, because that is not a defined alternative.

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.

Error: Search in BODY, HEAD or MESSAGE not `$in`.
The `in` option defines only three names.

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