

# mpop

---

A POP3 client  
version 0.6.3, May 29, 2005

Martin Lambers ([marlam@marlam.de](mailto:marlam@marlam.de))

---

This manual was last updated May 29, 2005 for version 0.6.3 of mpop.

Copyright (C) 2005 Martin Lambers

This program, including this manual, is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program, including this manual, is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>Configuration file .....</b>	<b>2</b>
2.1	General commands .....	2
2.2	Authentication commands .....	3
2.3	TLS commands .....	3
2.4	Commands specific to mail retrieval mode .....	4
<b>3</b>	<b>Invocation .....</b>	<b>7</b>
3.1	Synopsis .....	7
3.2	Exit code .....	7
3.3	Environment / Files .....	7
3.4	Options .....	7
3.4.1	General options .....	7
3.4.2	Changing the mode of operation .....	8
3.4.3	Configuration options .....	8
3.4.4	Options specific to mail retrieval mode .....	9
<b>4</b>	<b>POP3 features .....</b>	<b>11</b>
4.1	Transport Layer Security .....	11
4.2	Authentication .....	12
4.3	Pipelining .....	13
<b>5</b>	<b>Mail retrieval mode .....</b>	<b>14</b>
<b>6</b>	<b>Server information mode .....</b>	<b>15</b>
<b>7</b>	<b>Filtering .....</b>	<b>17</b>
<b>8</b>	<b>Examples .....</b>	<b>18</b>
8.1	A configuration file .....	18
8.2	Filtering with SpamAssassin .....	18
<b>9</b>	<b>Development .....</b>	<b>20</b>

# 1 Introduction

mpop is a POP3 client.

In its main mode of operation, it retrieves mails from one or more POP3 mailboxes, does some optional filtering, and delivers them through a mail delivery agent (MDA). Mails that were successfully delivered before can be identified and ignored. Successfully delivered mails will be deleted from POP3 server by default.

The best way to start is probably to have a look at the Examples section. See [Chapter 8 \[Examples\]](#), page 18.

In addition to the mail retrieval mode, mpop can be used in server information mode. In this mode, mpop prints as much information as it can get about a given POP3 server (greeting, supported features, login delay, maximum mail size, . . .).

Normally, a configuration file contains information about which POP3 server to use and how to use it, but almost all settings can also be configured on the command line.

POP3 server information is organized in accounts. Each account describes one POP3 server: host name, authentication settings, TLS settings, and so on. Each configuration file can define multiple accounts.

Supported features include:

- Authentication methods USER, APOP, PLAIN, LOGIN, CRAM-MD5 and EXTERNAL (and GSSAPI, DIGEST-MD5 and NTLM when compiled with GNU SASL support)
- TLS encrypted connections with the OpenSSL or GnuTLS libraries (including server certificate verification and the possibility to send a client certificate)
- POP3 command pipelining
- Header based mail filtering

## 2 Configuration file

If it exists and is readable, a user configuration file will be loaded (`$HOME/.mpoprc` by default). This file must have no more permissions than user read/write. Configuration file settings can be changed by command line options.

A configuration file is a simple text file. Empty lines and comment lines (whose first non-blank character is '#') are ignored. Every other line must contain a command and may contain an argument to that command. The argument may be enclosed in double quotes (").

If the first character of a filename is the tilde (~), this tilde will be replaced by `$HOME`.

If a command accepts the argument 'on', it also accepts an empty argument and treats that the same as 'on'.

Commands form groups. Each group starts with the 'account' command and defines the settings for one POP3 server.

See [Chapter 8 \[Examples\]](#), page 18.

### 2.1 General commands

'defaults'

Set defaults. The following configuration commands will set default values for all following account definitions.

'account *name* [ : *account* [, ... ] ]'

Start a new account definition with the given name. The current default values are filled in (see [\[defaults\]](#), page 2).

If a colon and a list of previously defined accounts is given after the account name, the new account, with the filled in default values, will inherit all settings from the accounts in the list.

'host *hostname*'

The POP3 server from which mails are to be retrieved. The argument may be a host name, an IPv4 address in dot notation, or (if your system supports it) an IPv6 address in colon notation. Every account definition must contain this command.

'port *number*'

The port that the POP3 server listens on. The default is 110, unless TLS without STARTTLS is used, in which case it is 995.

'connect\_timeout (*off*|*seconds*)'

Set or unset the connect timeout, in seconds. The argument off means that no timeout will be set, which means that the operating system default will be used.

'pipelining (*on*|*off*)'

Enable or disable POP3 pipelining. The default is off, because that works with every POP3 server. Most servers support pipelining, and using it usually makes POP3 sessions much faster, so it is recommended to activate it. See [Section 4.3 \[Pipelining\]](#), page 13.

`'pipeline_min n'`

Set the minimum length of the POP3 command pipeline. Default is 20. See [Section 4.3 \[Pipelining\]](#), page 13.

`'pipeline_max n'`

Set the maximum length of the POP3 command pipeline. Default is 100. See [Section 4.3 \[Pipelining\]](#), page 13.

## 2.2 Authentication commands

See [Section 4.2 \[Authentication\]](#), page 12.

`'auth [(on|method)]'`

This command chooses the POP3 authentication method. With the argument `on`, mpop will choose the best one available for you (see below). This is the default. Accepted methods are `user`, `apop`, `plain`, `cram-md5`, `digest-md5`, `gssapi`, `external`, `login`, and `ntlm`. See [Section 4.2 \[Authentication\]](#), page 12.

`'user [username]'`

Set your user name for POP3 authentication. An empty argument unsets the user name.

`'password [secret]'`

Set your password for POP3 authentication. An empty argument unsets the password. If no password is set but one is needed during authentication, mpop will try to find it in `~/.netrc`, and if that fails, mpop will prompt you for it. See [Section 4.2 \[Authentication\]](#), page 12.

`'ntlm domain [ntlm domain]'`

Set a domain for the `'ntlm'` authentication method. The default is to use no domain (equivalent to an empty argument), but some servers seem to require one, even if it is an arbitrary string.

## 2.3 TLS commands

See [Section 4.1 \[Transport Layer Security\]](#), page 11.

`'tls [(on|off)]'`

This command enables or disables TLS/SSL encrypted connections to the POP3 server. Not every server supports TLS, and many that do require the `'tls_starttls off'` command. See [Section 4.1 \[Transport Layer Security\]](#), page 11.

`'tls_trust_file [file]'`

This command activates strict server certificate verification. The file must contain one or more certificates of trusted Certification Authorities (CAs) in PEM format. An empty argument disables this feature. See [Section 4.1 \[Transport Layer Security\]](#), page 11.

`'tls_key_file [file]'`

This command (together with the `'tls_cert_file'`) command enables mpop to send a client certificate to the POP3 server if requested. The file must contain

the private key of a certificate in PEM format. An empty argument disables this feature. See [Section 4.1 \[Transport Layer Security\], page 11](#).

`'tls_cert_file [file]'`

This command (together with the `'tls_key_file'` command) enables mpop to send a client certificate to the POP3 server if requested. The file must contain a certificate in PEM format. An empty argument disables this feature. See [Section 4.1 \[Transport Layer Security\], page 11](#).

`'tls_certcheck [(on|off)]'`

This command enables or disables sanity checks for the server certificate. These checks are enabled by default, but can cause difficulties. See [Section 4.1 \[Transport Layer Security\], page 11](#).

`'tls_starttls [(on|off)]'`

This command enables or disables the use of the STARTTLS POP3 command to start TLS encryption. It is enabled by default. See [Section 4.1 \[Transport Layer Security\], page 11](#).

## 2.4 Commands specific to mail retrieval mode

See [Chapter 5 \[Mail retrieval mode\], page 14](#).

`'delivery method method_arguments...'`

How to deliver messages received from this account.

- `delivery mda command`  
Deliver the mails through a mail delivery agent (MDA). All occurrences of `%F` in the command will be replaced with the envelope from address of the current message (or MAILER-DAEMON if none is found). Use `delivery mda "/usr/bin/procmail -f %F -d $USER"` for the procmail MDA. Use `delivery mda "/usr/sbin/sendmail -oi -oem -f %F -- $USER"` to let your MTA handle the mail.
- `delivery mbox mbox-file`  
Deliver the mails to the given file in mbox format. The file will be locked with `fcntl(2)`. mpop uses the MBOXRD mbox format variant; see the documentation of the mbox format.  
Note that this method is not completely safe: In case of input/output errors, the last mail of the mbox file may be incomplete. This mail will not be deleted from the POP3 server, though, so you won't lose mail.
- `delivery maildir directory`  
Deliver the mails to the given maildir directory. The directory must exist and it must be a valid maildir directory; mpop will not create directories.
- `delivery simple_mbox mbox-file`  
This is the same as the mbox method, except that the mail headers are not parsed for the envelope from address. The `From_` lines will just contain a dash (-) instead. This is usually sufficient since the envelope from address information in `From_` lines is rarely used, and the mail still contains the relevant headers.

If the delivery method needs to parse the mail headers for an envelope from address (the mda method if the command contains %F, and the mbox method), then it needs to create a temporary file to store the mail headers (but not the body) in. See \$TMPDIR in [\[Environment / Files\]](#), page 7.

**‘uidls\_file filename’**

The file to store UIDs in. These are needed to identify new messages (see [\[only\\_new\]](#), page 5). The file is only used for POP3 servers that support the UIDL command; see [Chapter 6 \[Server information mode\]](#), page 15. The default is `~/.mpop_uidls`. One file can be shared by multiple accounts. An UIDs file contains at most as much UIDs as there are mails on the server, for each account it is used for.

**‘only\_new [(on|off)]’**

If this option is off, mpop processes all messages (this is the default). If it is on, mpop processes only new messages and ignores already retrieved messages. To distinguish between new and retrieved messages, it is necessary that the POP3 server supports the POP3 UIDL command (see [Chapter 6 \[Server information mode\]](#), page 15). For each retrieved mail that is left on the server, mpop stores a unique ID in a special file (see [\[uidls\\_file\]](#), page 5). When it runs the next time, it can recognize already retrieved mails because it knows their UIDs.

If the POP3 server does not support the UIDL command, all messages are new for mpop, so there’s no difference if this option is on or off for these servers.

This command is normally used in combination with the ‘keep’ command, see [\[keep\]](#), page 5.

**‘keep [(on|off)]’**

Keep all mails on the POP3 server, never delete them. The default behavior is to delete mails that have been successfully delivered or filtered by kill filters.

**‘killsize (off|size)’**

Mails larger than the given size will be deleted, not downloaded (unless the keep command is used, in which case they will just be skipped). The size argument must be zero or greater. If it is followed by a ‘k’ or an ‘m’, the size is measured in kilobytes/megabytes instead of bytes. Note that some POP3 servers report slightly incorrect sizes for mails. See [Chapter 7 \[Filtering\]](#), page 17.

**‘skipsize (off|size)’**

Mails larger than the given size will be skipped (not downloaded). The size argument must be zero or greater. If it is followed by a ‘k’ or an ‘m’, the size is measured in kilobytes/megabytes instead of bytes. Note that some POP3 servers report slightly incorrect sizes for mails. See [Chapter 7 \[Filtering\]](#), page 17.

**‘filter [COMMAND]’**

Set a filter which will decide whether to retrieve, skip, or delete each mail by investigating the mail’s headers. The POP3 server must support the POP3 TOP command for this to work; see [Chapter 6 \[Server information mode\]](#), page 15. An empty argument disables filtering.

All occurrences of %F in the command will be replaced with the envelope from address of the current message (or MAILER-DAEMON if none is found). All



occurrences of %S in the command will be replaced with the size of the current mail as reported by the POP3 server.

The mail headers (plus the blank line separating the headers from the body) will be piped to the command. Based on the return code, mpop decides what to do with the mail:

- 0: proceed normally; no special action
- 1: delete the mail; do not retrieve it
- 2: skip the mail; do not retrieve it

Return codes greater than or equal to 3 mean that an error occurred. The `sysexits.h` error codes may be used to give information about the kind of the error, but this is not necessary. See [Chapter 7 \[Filtering\]](#), page 17.

## 3 Invocation

### 3.1 Synopsis

- Mail retrieval mode (default):  
`mpop [option...] [--] [account...]`
- Server information mode:  
`mpop [option...] --serverinfo [account...]`

### 3.2 Exit code

The standard exit codes from `sysexits.h` are used.

### 3.3 Environment / Files

- `‘$HOME/.mpoprc’`  
 The default user configuration file.
- `‘$HOME/.mpop_uidls’`  
 Default file to store UIDLs in.
- `‘$HOME/.netrc’`  
 The `.netrc` file contains login information. If a password is not found in the configuration file, `msmtp` will search it in `.netrc` before prompting the user for it. The syntax of `.netrc` is described in the `netrc(5)` or `ftp(1)` manual page.
- `‘$USER, $LOGNAME’`  
 These variables override the user’s login name when writing mbox postmark lines with the mbox delivery method. `$LOGNAME` is only used if `$USER` is unset.
- `‘$TMPDIR’` Directory to create temporary files in. If this is unset, a system specific default directory is used.

### 3.4 Options

Options override configuration file settings. The following options are accepted:

#### 3.4.1 General options

- `‘--version’`  
 Print version information. This includes information about the library used for TLS/SSL support (if any), the library used for authentication, and the authentication mechanisms supported by this library.
- `‘--help’` Print help.
- `‘-p’`
- `‘--pretend’`  
 Print the configuration settings that would be used, but do not take further action. An asterisk (`‘*’`) will be printed instead of the password.

‘-d’

‘--debug’ Print lots of debugging information, including the whole conversation with the POP3 server. Be careful with this option: the (potentially dangerous) output will not be sanitized, and your password may get printed in an easily decodable format!

This option implies ‘--quiet’, because the debugging output would otherwise interfere with the normal output.

### 3.4.2 Changing the mode of operation

‘-S’

‘--serverinfo’

Print information about the POP3 server and exit. This includes information about supported features (authentication methods, TOP command, . . .), about parameters (time for which mails will not be deleted, minimum time between logins, . . .), and about the TLS certificate (if TLS is active). See [Chapter 6 \[Server information mode\]](#), page 15.

### 3.4.3 Configuration options

Most options in this category correspond to a configuration file command. Please refer to [Chapter 2 \[Configuration file\]](#), page 2 for detailed information.

`-C filename`  
`--file=filename`  
 Use the given file instead of `$HOME/.mpoprc` as the configuration file.

`--host=hostname`  
 Use this POP3 server with settings from the command line; do not use any configuration file data. This option disables loading of the configuration file. You cannot use both this option and account names on the command line.

`--port=number`  
 Set the port number to connect to. See [\[port\]](#), page 2.

`--connect-timeout=(off|seconds)`  
 Set or unset the connect timeout, in seconds. See [\[connect\\_timeout\]](#), page 2.

`--pipelining=(on|off)`  
 Enable or disable POP3 pipelining. See [\[pipelining\]](#), page 2.

`--pipeline-min=n`  
 Set the minimum length of the POP3 command pipeline. See [\[pipeline\\_min\]](#), page 2.

`--pipeline-max=n`  
 Set the maximum length of the POP3 command pipeline. See [\[pipeline\\_max\]](#), page 3.

`--auth[(on|method)]`  
 Set the authentication method to automatic (with "on") or manually choose an authentication method. See [\[auth\]](#), page 3.

`--user=[username]`  
 Set or unset the user name for authentication. See [\[user\]](#), page 3.

`--tls[(on|off)]`  
 Enable or disable TLS. See [\[tls\]](#), page 3.

`--tls-trust-file=[file]`  
 Set or unset a trust file for TLS encryption. See [\[tls\\_trust\\_file\]](#), page 3.

`--tls-key-file=[file]`  
 Set or unset a key file for TLS encryption. See [\[tls\\_key\\_file\]](#), page 3.

`--tls-cert-file=[file]`  
 Set or unset a cert file for TLS encryption. See [\[tls\\_cert\\_file\]](#), page 4.

`--tls-certcheck[(on|off)]`  
 Enable or disable server certificate checks for TLS encryption. See [\[tls\\_certcheck\]](#), page 4.

`--tls-starttls[(on|off)]`  
 Enable or disable STARTTLS for TLS encryption. See [\[tls\\_starttls\]](#), page 4.

### 3.4.4 Options specific to mail retrieval mode

`-q`  
`--quiet` Do not print progress information.

`-a`  
`--auth-only`  
 Authenticate only; do not retrieve mail. Useful for SMTP-after-POP.

`'-s'`  
`'--status-only'`  
Print number and size of mails in each account only; do not retrieve mail.

`'-n'`  
`'--only-new[=(on|off)]'`  
Process only new messages, ignore already retrieved messages. See [\[only\\_new\]](#), page 5.

`'-k'`  
`'--keep[=(on|off)]'`  
Do not delete mails from POP3 servers, regardless of other options or settings. See [\[keep\]](#), page 5.

`'--killsize=(off|size)'`  
Set or unset kill size. See [\[killsize\]](#), page 5.

`'--skipsize=(off|size)'`  
Set or unset skip size. See [\[skipsize\]](#), page 5.

`'--filter=[command]'`  
Set a filter which will decide whether to retrieve, skip, or delete each mail by investigating the mail's headers. See [\[filter\]](#), page 5.

`'--delivery=method,method_arguments...'`  
How to deliver messages received from this account. See [\[delivery\]](#), page 4. Note that a comma is used instead of a blank to separate the method from its arguments.

`'--uidls-file=filename'`  
File to store UIDs in. See [\[uidls\\_file\]](#), page 5.

## 4 POP3 features

### 4.1 Transport Layer Security

Quoting from RFC2246 - the TLS 1.0 protocol specification:

"The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery."

POP3 servers can use TLS in one of two modes:

- Immediately  
This is known as POP3 tunneled through TLS. The default port for this mode is 995 (pop3s).
- Via the STARTTLS POP3 command  
The POP3 session begins normally. The client sends the STLS command when it wishes to begin TLS encryption. The default port for this mode is the default POP3 port: 110 (pop3).

mpop can switch between these modes with the `'tls_starttls'` command (see [\[tls\\_starttls\]](#), page 4) command or the `'--tls-starttls'` option (see [\[-tls-starttls\]](#), page 9).

When TLS is started, the server sends a certificate to identify itself. This certificate contains information about the certificate owner, the certificate issuer, and the activation and expiration times of the certificate. This information can be displayed in server information mode. See [Chapter 6 \[Server information mode\]](#), page 15.

Some sanity checks are done with the server certificate. These include:

- Does the certificate belong to the host name of the POP3 server?
- Is the certificate activated?
- Is the certificate still valid, or has it expired?

Sometimes one of these checks fail. mpop will abort the connection in this case. If the user still wants to use this POP3 server with TLS, the sanity checks can be switched off with `'tls_certcheck'` or `'--tls-certcheck'` (see [\[tls\\_certcheck\]](#), page 4, [\[-tls-certcheck\]](#), page 9).

Note that the POP3 server cannot be fully trusted just because the certificate passes the sanity checks. To verify that the user can trust the POP3 server, it is necessary to use a (list of) certificates of Certification Authorities (CAs) that are trusted. If mpop can verify that the server certificate was issued by one of these CAs, then the POP3 server is trusted. A file containing CA certificates can be set with `'tls_trust_file'` or `'--tls-trust-file'` (see [\[tls\\_trust\\_file\]](#), page 3, [\[-tls-trust-file\]](#), page 9).

If the server requests it, the client can send a certificate, too. This allows the server to verify the identity of the client. See the EXTERNAL mechanism in [Section 4.2 \[Authentication\]](#), page 12. The `'tls_key_file'`/`'tls_cert_file'` commands or the `'--tls-key-file'`/`'--tls-cert-file'` options can be used to set a client certificate. See [\[tls\\_key\\_file\]](#), page 3/[\[-tls-key-file\]](#), page 9, [\[tls\\_cert\\_file\]](#), page 4/[\[-tls-cert-file\]](#), page 9. Note that GnuTLS will only send a client certificate if it matches one of the CAs advertised by the server. If you set a client certificate but it is not send to the server, probably does not match any CA that the server trusts.

## 4.2 Authentication

POP3 servers require a client to authenticate itself before it is allowed to retrieve mail.

Multiple authentication methods exist. Most POP3 servers support only some of them. Some methods send authentication data in plain text (or nearly plain text) to the server. These methods should only be used when TLS is active to prevent others from stealing the password. See [Section 4.1 \[Transport Layer Security\], page 11](#).

mpop supports a subset of the following authentication methods:

- ‘USER’  
This authentication method needs a user name and a password. Both are send in plain text. All POP3 servers support this authentication method.
- ‘APOP’  
This authentication method needs a user name and a password. The authentication data is not sent in plain text, which means this method can safely be used without TLS.
- ‘PLAIN’  
This authentication method needs a user name and a password. Both are send in BASE64 encoding, which can be easily decoded to plain text.
- ‘CRAM-MD5’  
This authentication method needs a user name and a password. The authentication data is not sent in plain text, which means this method can safely be used without TLS.
- ‘DIGEST-MD5’  
This authentication method needs a user name and a password. The authentication data is not sent in plain text, which means this method can safely be used without TLS.
- ‘GSSAPI’  
This authentication method needs a user name. The Kerberos framework takes care of secure authentication, therefore this method can be used without TLS.
- ‘EXTERNAL’  
This is a special authentication method: The actual authentication happens outside of the POP3 protocol, typically by sending a TLS client certificate (see [Section 4.1 \[Transport Layer Security\], page 11](#)).  
The EXTERNAL method merely confirms that this authentication succeeded for the given user (or, if no user name is given, confirms that authentication succeeded). Thus it may not be necessary for authentication to use this method, and if the server does not support the EXTERNAL method, this does not mean that it does not support authentication with TLS client certificates.  
This authentication method is not chosen automatically; you have to request it manually.  
Note: (SMTP) Sendmail 8.12.11 advertises the EXTERNAL mechanism only after a TLS client certificate has been send. It seems to ignore the optional user name. Does anyone know more about this?
- ‘LOGIN’  
This is a non-standard authentication method similar to (but worse than) PLAIN. It

needs a user name and a password, both of which are send in BASE64 encoding, which can be easily decoded to plain text.

- ‘NTLM’

This is a obscure non-standard authentication method. It needs a user name and a password and in some cases a special domain parameter (see [\[ntlm domain\]](#), page 3). The authentication data is not send in plain text.

It depends on the underlying authentication library and its version whether a particular method is supported or not. Use the ‘--version’ to find out which methods are supported by your version of mpop.

Authentication data can be set with the ‘user’ and ‘password’ commands or with the ‘--user’ option. See [\[user\]](#), page 3, [\[password\]](#), page 3, [\[-user\]](#), page 9. If no password is set but one is needed during authentication, mpop will try to find it in `~/.netrc`, and if that fails, mpop will prompt you for it.

The authentication method can be chosen with the ‘auth’ command or ‘--auth’ option, but it is usually sufficient to just use the ‘on’ argument to let mpop choose the method itself. See [\[auth\]](#), page 3, [\[-auth\]](#), page 9.

If mpop chooses the method itself, it will not choose a method that sends plain text authentication data when TLS is not active. This means that only APOP, CRAM-MD5, DIGEST-MD5, GSSAPI, and NTLM are available when TLS is inactive. USER, PLAIN and LOGIN are only available when TLS is active. If you really want to send clear text authentication data, you have to force mpop to do that by setting the authentication method to USER, PLAIN or LOGIN while TLS is off.

## 4.3 Pipelining

Pipelining can be enabled with the ‘**pipelining**’ command or the ‘--pipelining’ option. In short: enable it, it will speed up the POP3 sessions.

A POP3 client that sends multiple POP3 commands at once to a POP3 server before starting to read the server’s answers is using POP3 pipelining. Since the client does not have to wait for the server’s answer before sending the next command, and the server does not have to wait for the next command from the client, pipelining can speed up a POP3 session substantially.

Most POP3 servers support pipelining. Those that support the CAPA command, too, will advertize the pipelining capability; see [Chapter 6 \[Server information mode\]](#), page 15. But even if a server does not support CAPA, it probably still supports pipelining.

Pipelining in mpop works by sending up to ‘**pipeline\_max**’ commands to the server, then begin to read its answers, and refill the command pipeline when the number of unanswered commands drops to ‘**pipeline\_min**’. Both parameters can be changed with the ‘**pipeline\_min**’/‘**pipeline\_max**’ command and the corresponding options, but this is not necessary; the defaults should be good enough.

See [\[pipelining\]](#), page 2, [\[-pipelining\]](#), page 9.



## 5 Mail retrieval mode

In this mode, mpop retrieves mail from one or more POP3 servers. It delivers each of them using the method that was given with the ‘`delivery`’ command or ‘`--delivery`’ option. See [\[delivery\]](#), page 4, [\[-delivery\]](#), page 10.

While retrieving the mail, mpop displays approximate progress information, which can be turned off with the ‘`--quiet`’ option; see [\[-quiet\]](#), page 9.

If the delivery succeeded, the mail is deleted from the POP3 server by default. The ‘`keep`’ command and ‘`--keep`’ option can prevent the deletion of mails; see [\[keep\]](#), page 5, [\[-keep\]](#), page 10.

*Important note:* Some POP3 servers will delete mails without any user interaction. See EXPIRE in [Chapter 6 \[Server information mode\]](#), page 15. mpop can do nothing about that.

If you don’t want to download certain mails, but skip them or delete them directly, you can do filtering based on the mail headers. See [Chapter 7 \[Filtering\]](#), page 17.

If you just want to know if you have new mails (and if so, how many), use the ‘`--status-only`’ option. See [\[-status-only\]](#), page 9.

If you just want to authenticate to the POP3 server, but don’t want to look at your mails, use the ‘`--auth-only`’ option. See [\[-auth-only\]](#), page 9. This can be useful for sending mail through SMTP servers that require SMTP-after-POP (aka POP-before-SMTP).

## 6 Server information mode

In server information mode, mpop prints as much information about the POP3 server as it can get and then exits.

The POP3 features that can be detected are:

- **IMPLEMENTATION**  
The implementation string of the POP3 server.
- **CAPA**  
Support for the POP3 CAPA command. The server sends a list of its capabilities in response to this command.
- **PIPELINING**  
Support for POP3 pipelining. See [Section 4.3 \[Pipelining\]](#), page 13. Most POP3 servers support this, even those that do not support the CAPA command and thus do not advertize the PIPELINING capability.
- **TOP**  
Support for the POP3 TOP command. This is needed for header based filtering to work. See [Chapter 7 \[Filtering\]](#), page 17.
- **UIDL**  
Support for the POP3 UIDL command. This is needed to distinguish between new and already retrieved messages. See [\[only-new\]](#), page 5.
- **LOGIN-DELAY**  
The minimum time between two POP3 sessions. The server may refuse a POP3 session if the last one was active less than this time period ago.
- **EXPIRE**  
The time after which old mails are deleted by the POP3 server.
  - **NEVER**: The POP3 server will not delete mail without the user requesting it.
  - **0**: The POP3 server will not keep mails; all mails will be deleted after they have been downloaded, regardless of the user's wishes.
  - *number*: The number of days that the POP3 server will keep mails before deleting them without user interaction.
- **STARTTLS**  
See [Section 4.1 \[Transport Layer Security\]](#), page 11.
- **AUTH**  
See [Section 4.2 \[Authentication\]](#), page 12.
- **RESP-CODES**  
If authentication fails and the POP3 server issues an error message beginning with a square bracket, this message will include additional information about the source of the error:
  - **[LOGIN-DELAY]**: The login delay period has not yet expired.
  - **[IN-USE]**: Authentication succeeded but the mailbox is currently in use, possibly by another POP3 session.
- **AUTH-RESP-CODE**  
If authentication fails and the POP3 server issues an error message beginning with a

square bracket, this message will include additional information about the source of the error:

- [LOGIN-DELAY]: The login delay period has not yet expired.
- [IN-USE]: Authentication succeeded but the mailbox is currently in use, possibly by another POP3 session.
- [SYS/TEMP]: Temporary system failure; try again later.
- [SYS/PERM]: Permanent system failure; ask the administrator.
- [AUTH]: Incorrect user name or password or some other problem with the user's credentials.

If TLS is activated for server information mode, the following information will be printed about the POP3 server's TLS certificate (if available):

- Owner information
  - Common Name
  - Organization
  - Organizational unit
  - Locality
  - State or Province
  - Country
- Issuer information
  - Common Name
  - Organization
  - Organizational unit
  - Locality
  - State or Province
  - Country
- General
  - Activation time
  - Expiration time
  - SHA1 fingerprint
  - MD5 fingerprint

## 7 Filtering

There are three filtering commands available. They will be executed in the following order:

1. ‘killsize’
2. ‘skipsize’
3. ‘filter’

If a filtering command applies to a mail, the remaining filters will not be executed.

The POP3 server must support the POP3 TOP command ([Chapter 6 \[Server information mode\], page 15](#)) for filtering with a filter command: It is used to read the mail headers (plus the blank line separating the header from the body) and pipe them to the filter command.

Note that, if the filter decides that the mail should be retrieved, the complete mail has to be downloaded, including the headers, so the headers will be downloaded twice. This is because there’s no way in POP3 to download just the mail body. Sometimes this overhead surpasses the savings of the filtering.

The filter command looks at the mail headers and signals with its exit code what mpop should do with the mail:

- 0: retrieve the mail
- 1: delete the mail; do not retrieve it
- 2: skip the mail; do not retrieve it

Return codes greater than or equal to 3 mean that an error occurred. The `sysexits.h` error codes may be used to give information about the kind of the error, but this is optional.

Since the filter command will be passed to a shell, you can use all shell command constructs in addition to just calling a script or program. This allows flexible filter constructs. See [Section 8.2 \[Filtering with SpamAssassin\], page 18](#).

Some POP3 servers count end-of-line characters as two bytes (CRLF) instead of one (LF), so that the size of a mail as reported by the POP3 server is slightly larger than the actual size. The filters use the size values reported by the POP3 server since they cannot know the actual size in advance. Thus you cannot rely on *exact* size filtering.

## 8 Examples

### 8.1 A configuration file

```
# Default values for all accounts
defaults
pipelining on
delivery mda "/usr/bin/procmail -f %F -d $USER"
#delivery mda "/usr/sbin/sendmail -oi -oem -f %F -- $USER"
#delivery maildir ~/Mail/incoming
#delivery mbox ~/Mail/new
#delivery simple_mbox ~/Mail/new
tls on
tls_trust_file ~/.ca-bundle.pem

# My provider, where I have two pop3 mailboxes
account provider1
host mx.provider.example
user joe_smith
password secret
tls_starttls off

account provider2 : provider1
user joey
password secret2

# A freemail service
account freemail
host pop.freemail.example
user 123
password pass

# Be sure to never loose mail while testing a new
# filter script for the freemail service
account freemail-test : freemail
filter ~/bin/experimental-filter.sh --from=%F --size=%S
keep on

# Set a default account
account default : provider1
```

### 8.2 Filtering with SpamAssassin

Use the following to delete all mails that SpamAssassin classifies as spam:

```
filter "/path/to/spamc -c > /dev/null"
```

Since no message body is passed to SpamAssassin, you should disable all body-specific tests in the SpamAssassin configuration file; for example set `use_bayes 0`.

If your mail provider runs SpamAssassin for you, you just have to check for the result. The following script can do that when used as an mpop filter:

```
#!/bin/sh
if [ "$(grep "^X-Spam-Status: Yes" )" ]; then
    exit 1 # kill this message
else
    exit 0 # proceed normally
fi
```

Since the filter command is passed to a shell, all shell constructs are usable, so you can also use this directly:

```
filter if [ "$(grep "^X-Spam-Status: Yes" )" ]; then exit 1; else exit 0; fi
```

## 9 Development

The homepage of mpop is <http://mpop.sourceforge.net/>; the SourceForge project page is at <http://sourceforge.net/projects/mpop/>.

Please send any questions, suggestions, and bug reports to Martin Lambers ([marlam@marlam.de](mailto:marlam@marlam.de), OpenPGP key: <http://www.marlam.de/key.txt>). If you send a bug report, please include the output of `mpop --version`.