msmtp

Version 1.4.31, 22 April 2013

Martin Lambers (marlam@marlam.de)

This manual was last updated 22 April 2013 for version 1.4.31 of msmtp.

Table of Contents

1 Introduction

msmtp is an SMTP client.

In its default mode of operation, it reads a mail from standard input and sends it to a predefined SMTP server that takes care of proper delivery. Command line options and exit codes are compatible to sendmail.

Supported SMTP features include:

- Authentication methods PLAIN, LOGIN, CRAM-MD5 and EXTERNAL (and GSS-API, SCRAM-SHA-1, 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)
- Support for Internationalized Domain Names (IDN)
- DSN (Delivery Status Notification) support
- PIPELINING support for increased transmission speed
- RMQS (Remote Message Queue Starting) support (ETRN keyword)

The best way to start is probably to have a look at the Examples section. See Chapter 10 [Examples], page 22.

In addition to sendmail mode, there are two other modes of operation:

- Server information mode. In this mode, msmtp prints as much information as it can get about a given SMTP server (supported features, maximum mail size, . . .).
- Remote Message Queue Starting mode. In this mode, msmtp sends a Remote Message Queue Starting request for a host, domain, or queue to a given SMTP server.

Normally, a system wide configuration file and/or a user configuration file contain information about which SMTP server to use and how to use it, but almost all settings can also be configured on the command line.

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

2 Configuration files

msmtp supports a system wide configuration file and a user configuration file. Both are optional and need not exist.

If it exists and is readable, a system wide configuration file `SYSCONFDIR/msmtprc` will be loaded, where `SYSCONFDIR` depends on your platform. The default is `/usr/local/etc`. Use `--version` to find out which directory your version uses.

If it exists and is readable, a user configuration file will be loaded (`~/.msmtprc` by default). Accounts defined in the user configuration file override accounts from the system configuration file. The user configuration file must have no more permissions than user read/write. Configuration data from either file 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 as if it was '`on`'.

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

See Chapter 10 [Examples], page 22.

2.1 General commands

'`defaults`'
: Set defaults. The following configuration commands will set default values for all following account definitions in the current configuration file.

'`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 SMTP server to send the mail to. The argument may be a host name or a network address. Every account definition must contain this command.

'`port` *number*'
: The port that the SMTP server listens on. The default port will be acquired from your operating system's service database: for SMTP, the service is "smtp" (default port 25), unless TLS without STARTTLS is used, in which case it is "smtps" (465). For LMTP, it is "lmtp".

'`timeout (off|`*seconds*`)`'
: Set or unset a network timeout, in seconds. The argument '`off`' means that no timeout will be set, which means that the operating system default will be

used. For compatibility with older versions, '`connect_timeout`' is accepted as an alias for this command.

'`protocol (smtp|lmtp)`'
: Set the protocol to use. Currently only SMTP and LMTP are supported. SMTP is the default. See [port], page 2 for default ports.

'`domain` *argument*'
: This command sets the argument of the SMTP EHLO (or LMTP LHLO) command. The default is '`localhost`', which is stupid but usually works. Try to change the default if mails get rejected due to anti-SPAM measures. Possible choices are the domain part of your mail address (`provider.example` for `joe@provider.example`) or the fully qualified domain name of your host (if available).

2.2 Authentication commands

See Chapter 5 [Authentication], page 15.

'`auth [(on|off|`*method*`)]`'
: This command enables or disables SMTP authentication and optionally chooses an authentication method to use. It should not be necessary to choose a method; with the argument '`on`', msmtp will choose the best one available. Accepted methods are '`plain`', '`scram-sha-1`', '`cram-md5`', '`gssapi`', '`external`', '`digest-md5`', '`login`', and '`ntlm`'. See Chapter 5 [Authentication], page 15.

'`user [`*username*`]`'
: Set your user name for SMTP authentication. An empty argument unsets the user name. Authentication must be activated with the '`auth`' command.

'`password [`*secret*`]`'
: Set your password for SMTP authentication. An empty argument unsets the password. Authentication must be activated with the '`auth`' command. If no password is set but one is needed during authentication, msmtp will try to find it. First, if '`passwordeval`' is set, it will evaluate that command. If '`passwordeval`' is not set, msmtp will try to find the password in `~/.netrc`. If that fails, it will try to find it in `SYSCONFDIR/netrc` (use `--version` to find out what `SYSCONFDIR` is on your platform). If that fails, it will try to get it from a system specific keyring (if available). If that fails but a controlling terminal is available, msmtp will prompt you for it. See Chapter 5 [Authentication], page 15.

'`passwordeval [`*eval*`]`'
: Set your password for SMTP authentication to the output (stdout) of the execution of *eval*.

'`ntlmdomain [`*ntlmdomain*`]`'
: 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 Chapter 4 [Transport Layer Security], page 13.

'`tls [(on|off)]`'
: This command enables or disables TLS/SSL encrypted connections to the SMTP server. Not every server supports TLS, and a few that support it require the '`tls_starttls off`' command.
To use TLS/SSL, it is required to either use the '`tls_trust_file`' command (highly recommended) or to disable '`tls_certcheck`'. See Chapter 4 [Transport Layer Security], page 13.

'`tls_trust_file [file]`'
: This command activates strict server certificate verification. The given file must contain one or more certificates of trusted Certification Authorities (CAs) in PEM format.
On Debian based systems, you can install the '`ca-certificates`' package and use the file '`/etc/ssl/certs/ca-certificates.crt`'.
An empty argument disables this feature.

'`tls_crl_file [file]`'
: This command sets or unsets a certificate revocation list (CRL) file for TLS, to be used during strict server certificate verification as enabled by the [tls_trust_file], page 4 command. This allows the verification procedure to detect revoked certificates. See Chapter 4 [Transport Layer Security], page 13.

'`tls_fingerprint [fingerprint]`'
: This command sets or unsets the fingerprint of a particular TLS certificate. This certificate will then be trusted, regardless of its contents. This can be used to trust broken certificates (e.g. with a non-matching hostname) or in situations where '`tls_trust_file`' cannot be used for some reason. You can give either an SHA1 (recommended) or an MD5 fingerprint in the format `01:23:45:67:...`.
You can use '`--serverinfo --tls --tls-certcheck=off`' to get the peer certificate's fingerprints. See Chapter 4 [Transport Layer Security], page 13.

'`tls_key_file [file]`'
: This command (together with the '`tls_cert_file`') command enables msmtp to send a client certificate to the SMTP server if requested. The file must contain the private key of a certificate in PEM format. An empty argument disables this feature. See Chapter 4 [Transport Layer Security], page 13.

'`tls_cert_file [file]`'
: This command (together with the '`tls_key_file`' command) enables msmtp to send a client certificate to the SMTP server if requested. The file must contain a certificate in PEM format. An empty argument disables this feature. See Chapter 4 [Transport Layer Security], page 13.

'`tls_certcheck [(on|off)]`'
: This command enables or disables checks for the server certificate.
WARNING: When the checks are disabled, TLS/SSL sessions will be vulnerable to man-in-the-middle attacks! See Chapter 4 [Transport Layer Security],

page 13.
For compatibility with older versions, '`tls_nocertcheck`' is accepted as an alias for '`tls_certcheck off`'.

'`tls_starttls [(on|off)]`'
This command enables or disables the use of the STARTTLS SMTP command to start TLS encryption. It is enabled by default. See Chapter 4 [Transport Layer Security], page 13. For compatibility with older versions, '`tls_nostarttls`' is accepted as an alias for '`tls_starttls off`'.

'`tls_force_sslv3 [(on|off)]`'
Force TLS/SSL version SSLv3. This might be needed to use SSL with some old and broken servers. Do not use this unless you have to. See Chapter 4 [Transport Layer Security], page 13.

'`tls_min_dh_prime_bits [`*bits*`]`'
Set or unset the minimum number of Diffie-Hellman (DH) prime bits that msmtp will accept for TLS sessions. The default is set by the TLS library and can be selected by using an empty argument to this command. Only lower the default (for example to 512 bits) if there is no other way to make TLS work with the remote server. See Chapter 4 [Transport Layer Security], page 13.

'`tls_priorities [`*priorities*`]`'
Set the priorities for TLS sessions. The default is set by the TLS library and can be selected by using an empty argument to this command. Currently this command only works with sufficiently recent GnuTLS releases. See the GnuTLS documentation of the '`gnutls_priority_init`' function for a description of the *priorities* string. See Chapter 4 [Transport Layer Security], page 13.

2.4 Commands specific to sendmail mode

See Chapter 7 [Sendmail mode], page 18.

'`auto_from [(on|off)]`'
Enable or disable automatic envelope-from addresses. The default is '`off`'.
When enabled, an envelope-from address of the form user@domain will be generated. The local part will be set to `USER` or, if that fails, to `LOGNAME` or, if that fails, to the login name of the current user. The domain part can be set with the '`maildomain`' command (see [maildomain], page 5). If the maildomain is empty, the envelope-from address will only consist of the user name and not have a domain part.
When disabled, the envelope-from address must be set explicitly with the '`from`' command (see [from], page 5).
See Section 7.1 [Envelope-from address], page 18.

'`from [`*address*`]`'
Set the envelope-from address. This address will only be used when '`auto_from`' is disabled. See Section 7.1 [Envelope-from address], page 18.

'`maildomain [domain]`'
: Set a domain part for the generation of an envelope-from address. This is only used when '`auto_from`' is enabled. The domain may be empty. See Section 7.1 [Envelope-from address], page 18.

'`dsn_notify (off|condition)`'
: This command sets the condition(s) under which the mail system should send DSN (Delivery Status Notification) messages. The argument off disables explicit DSN requests, which means the mail system decides when to send DSN messages. This is the default. The *condition* must be '`never`', to never request notification, or a comma separated list (no spaces!) of one or more of the following: '`failure`', to request notification on transmission failure, '`delay`', to be notified of message delays, '`success`', to be notified of successful transmission. The SMTP server must support the DSN extension. See Chapter 6 [Delivery Status Notifications], page 17.

'`dsn_return (off|amount)`'
: This command controls how much of a mail should be returned in DSN (Delivery Status Notification) messages. The argument off disables explicit DSN requests, which means the mail system decides how much of a mail it returns in DSN messages. This is the default. The *amount* must be '`headers`', to just return the message headers, or '`full`', to return the full mail. The SMTP server must support the DSN extension. See Chapter 6 [Delivery Status Notifications], page 17.

'`keepbcc [(on|off)]`'
: This command controls whether to remove or keep the Bcc header when sending a mail. The default is to remove it. See Section 7.3 [Bcc header], page 19.

'`logfile [file]`'
: This command enables or disables logging to the specified file. An empty argument disables this feature. The file name '`-`' directs the log information to standard output. See Section 7.2 [Logging], page 18.

'`syslog [(on|off|facility)]`'
: This command enables or disables syslog logging. The facility can be one of '`LOG_USER`', '`LOG_MAIL`', '`LOG_LOCAL0`', ..., '`LOG_LOCAL7`'. The default facility is '`LOG_USER`'. Syslog logging is disabled by default. See Section 7.2 [Logging], page 18.

'`aliases [file]`'
: Replace local recipients with addresses in the aliases file. The aliases file is a plain text file containing mappings between a local address and a list of domain addresses. A local address is defined as one without an '`@`' character and a domain address is one with an '`@`' character. The mappings are of the form:

```
local: someone@example.com, person@domain.example
```

Multiple domain addresses are separated with commas. Comments start with '#' and continue to the end of the line.
The local address '`default`' has special significance and is matched if the local address is not found in the aliases file. If no '`default`' alias is found, then the

local address is left as is.
An empty argument to the aliases command disables the replacement of local addresses. This is the default.

3 Invocation

3.1 Synopsis

- Sendmail mode (default):
  `msmtp [`*`option`*`...] [--] `*`recipient`*`...`
  `msmtp [`*`option`*`...] -t [--] [`*`recipient`*`...]`
- Server information mode:
  `msmtp [`*`option`*`...] --serverinfo`
- Remote Message Queue Starting mode:
  `msmtp [`*`option`*`...] --rmqs=(`*`host`*`|@`*`domain`*`|#`*`queue`*`)`

3.2 Options

Options override configuration file settings. They are compatible with sendmail where appropriate.

3.2.1 General options

'`--version`'
: Print version information. This includes information about the library used for TLS/SSL support (if any), the library used for authentication, the authentication mechanisms supported by this library, and the default locations of the system and user configuration files.

'`--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.

'`-v`'

'`-d`'
'`--debug`'
: Print lots of debugging information, including the whole conversation with the SMTP 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!

3.2.2 Changing the mode of operation

'`-S`'
'`--serverinfo`'
: Print information about the SMTP server and exit. This includes information about supported features (mail size limit, authentication, TLS, DSN, ...) and about the TLS certificate (if TLS is active). See Chapter 8 [Server information mode], page 20.

‘`--rmqs=(host|@domain|#queue)`’
: Send a Remote Message Queue Starting request for the given host, domain, or queue to the SMTP server and exit. See Chapter 9 [Remote Message Queue Starting mode], page 21.

3.2.3 Configuration options

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

‘`-C filename`’
‘`--file=filename`’
: Use the given file instead of `~/.msmtprc` as the user configuration file.

‘`-a account`’
‘`--account=account`’
: Use the given account instead of the account named ‘`default`’. This option cannot be used together with the ‘`--host`’ option. See [Choosing an account], page 11.

‘`--host=hostname`’
: Use this SMTP server with settings from the command line; do not use any configuration file data. This option cannot be used together with the ‘`--account`’ option. It disables loading of configuration files. See [Choosing an account], page 11.

‘`--port=number`’
: Set the port number to connect to. See [port], page 2.

‘`--timeout=(off|seconds)`’
: Set a network timeout. See [timeout], page 2. For compatibility with older versions, ‘`--connect-timeout`’ is accepted as an alias for this option.

‘`--protocol=(smtp|lmtp)`’
: Set the protocol. See [protocol], page 3.

‘`--domain=[argument]`’
: Set the argument of the SMTP EHLO (or LMTP LHLO) command. See [domain], page 3.

‘`--auth[=(on|off|method)]`’
: Enable or disable authentication and optionally choose the method. See [auth], page 3.

‘`--user=[username]`’
: Set or unset the user name for authentication. See [user], page 3.

‘`--passwordeval=[eval]`’
: Evaluate password for authentication. See [passwordeval], page 3.

‘`--tls[=(on|off)]`’
: Enable or disable TLS/SSL. See [tls], page 4.

‘`--tls-starttls[=(on|off)]`’
: Enable or disable STARTTLS for TLS encryption. See [tls_starttls], page 5.

‘`--tls-trust-file=[file]`’
: Set or unset a trust file for TLS encryption. See [tls_trust_file], page 4.

‘`--tls-crl-file=[file]`’
: Set or unset a certificate revocation list (CRL) file for TLS. See [tls_crl_file], page 4.

‘`--tls-fingerprint=[fingerprint]`’
: Set ot unset the fingerprint of a trusted TLS certificate. See [tls_fingerprint], page 4.

‘`--tls-key-file=[file]`’
: Set or unset a key file for TLS encryption. See [tls_key_file], page 4.

‘`--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-force-sslv3[=(on|off)]`’
: Force TLS/SSL version SSLv3. See [tls_force_sslv3], page 5.

‘`--tls-min-dh-prime-bits=[bits]`’
: Set or unset minimum bit size of the Diffie-Hellman (DH) prime. See [tls_min_dh_prime_bits], page 5.

‘`--tls-priorities=[priorities]`’
: Set or unset TLS priorities. See [tls_priorities], page 5.

3.2.4 Options specific to sendmail mode

‘`--auto-from[=(on|off)]`’
: Enable or disable automatic envelope-from addresses. The default is off. See [auto_from], page 5.

‘`-f address`’
‘`--from=address`’
: Set the envelope-from address. It is only used when ‘`auto_from`’ is off. See [from], page 5.
If no account was chosen yet (with ‘`--account`’ or ‘`--host`’), this option will choose the first account that has the given envelope-from address (set with the ‘`from`’ command). If no such account is found, "default" is used. See [Choosing an account], page 11.

‘`--maildomain=[domain]`’
: Set the domain part for generated envelope-from addresses. It is only used when ‘`auto_from`’ is on. See [maildomain], page 5.

‘`-N (off|condition)`’
‘`--dsn-notify=(off|condition)`’
: Set or unset DSN notification conditions. See [dsn_notify], page 6.

‘`-R (off|amount)`’
‘`--dsn-return=(off|amount)`’
: Set or unset the DSN notification amount. See [dsn_return], page 6. Note that ‘`hdrs`’ is accepted as an alias for ‘`headers`’ to be compatible with sendmail.

‘`--keepbcc[=(on|off)]`’
: Enable or disable the preservation of the Bcc header. See [keepbcc], page 6.

‘`-X [file]`’
‘`--logfile=[file]`’
: Set or unset the log file. See [logfile], page 6.

‘`--syslog[=(on|off|facility)]`’
: Enable or disable syslog logging. See [syslog], page 6.

‘`-t`’
‘`--read-recipients`’
: Send the mail to the recipients given in the To, Cc, and Bcc headers of the mail in addition to the recipients given on the command line.
If any Resent- headers are present, then the addresses from any Resent-To, Resent-Cc, and Resent-Bcc headers in the first block of Resent- headers are used instead.

‘`--read-envelope-from`’
: Read the envelope from address from the From header of the mail. Currently this header must be on a single line for this option to work correctly.

‘`--aliases=[file]`’
: Set or unset an aliases file. See [aliases], page 6.

‘`--`’
: This marks the end of options. All following arguments will be treated as recipient addresses, even if they start with a ’-’.

The following options are accepted but ignored for sendmail compatibility: ‘`-Btype`’, ‘`-bm`’, ‘`-Fname`’, ‘`-G`’, ‘`-hN`’, ‘`-i`’, ‘`-L tag`’, ‘`-m`’, ‘`-n`’, ‘`-O option=value`’, ‘`-ox value`’

3.3 Choosing an account

There are three ways to choose the account to use. It depends on the circumstances which method is the best.

1. ‘`--account=account`’
   Use the given account. Command line settings override configuration file settings.
2. ‘`--host=hostname`’
   Use only the settings from the command line; do not use any configuration file data.
3. ‘`--from=address`’ or ‘`--read-envelope-from`’
   Choose the first account from the system or user configuration file that has a matching envelope-from address as specified by a ‘`from`’ command. This works only when neither ‘`--account`’ nor ‘`--host`’ is used.

If none of the above options is used (or if no account has a matching ‘`from`’ command), then the account "default" is used.

3.4 Exit code

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

3.5 Files

'`SYSCONFDIR/msmtprc`'
: The system configuration file. Use the '`--version`' option to find out what `SYSCONFDIR` is on your platform.

'`~/.msmtprc`'
: The default user configuration file.

'`~/.netrc and SYSCONFDIR/netrc`'
: The `netrc` file contains login information. If a password is not found in the configuration file, msmtp will search it in `~/.netrc` and `SYSCONFDIR` before prompting the user for it. The syntax of `netrc` files is described in the `netrc(5)` or `ftp(1)` manual page.

3.6 Environment

'`USER, LOGNAME`'
: These variables override the user's login name when constructing an envelope-from address. `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.
A temporary file is only created when the '`-t`'/'`--read-recipients`' or '`--read-envelope-from`' option is used. The file is then used to buffer the headers of the mail (but not the body, so the file won't get very large).

'`EMAIL, SMTPSERVER`'
: These environment variables are used only if neither '`--host`' nor '`--account`' is used and there is no default account defined in the configuration files. In this case, the host name is taken from `SMTPSERVER`, and the envelope from address is taken from `EMAIL`, unless overridden by '`--from`' or '`--read-envelope-from`'. Currently `SMTPSERVER` must contain a plain host name (no URL), and `EMAIL` must contain a plain address (no names or additional information).

4 Transport Layer Security

Transport Layer Security (TLS) is a new name for Secure Socket Layer (SSL). The TLS 1.0 protocol is an updated version of the SSL 3.0 protocol. TLS and SSL mean the same thing.

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."

SMTP servers can use TLS in one of two modes:

- Immediately. This is SMTP tunneled through TLS, aka SSMTP. The default port for this mode is 465 (smtps).
- Via the STARTTLS SMTP command. The SMTP session begins normally. The client sends the STARTTLS command when it wishes to begin TLS encryption. The default port for this mode is the default SMTP port: 25 (smtp).

msmtp can switch between these modes with the '`tls_starttls`' command (see [tls_starttls], page 5) 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 8 [Server information mode], page 20.

To use TLS, it is required to either enable full server certificate verification using the '`tls_trust_file`' command or '`--tls-trust-file`' option, or to trust one particular peer certificate using the '`tls_fingerprint`' command or '`--tls-fingerprint`' option, or to disable all certificate checks using '`tls_certcheck off`' or '`--tls-certcheck=off`'. WARNING: When certificate checks are disabled, TLS/SSL sessions are vulnerable to man-in-the-middle attacks! See [tls_trust_file], page 4, [–tls-trust-file], page 9, [tls_fingerprint], page 4, [–tls-fingerprint], page 10, [tls_certcheck], page 4, [–tls-certcheck], page 10.

If your system has a file that collects all system-wide trusted CA certificates, it is easiest to just use this in the '`defaults`' section of your configuration file. On Debian-based systems, for example, the adequate command would be '`tls_trust_file /etc/ssl/certs/ca-certificates.crt`'.

But you can also find out manually which CA certificate you need to trust. First, issue the following command:

```
$ msmtp --serverinfo --host=smtp.example.com --tls=on --tls-certcheck=off
```

The option '`--tls-certcheck=off`' allows msmtp to accept any certificate, so that it can print some information about it. The output of this command tells you the common name of the server certificate issuer. You have to trust this issuer to use full TLS security. Usually you can find the CA certificate on the issuer's homepage. With this CA certificate, the following should succeed:

```
$ msmtp --serverinfo --host=smtp.example.com --tls=on \
  --tls-trust-file=ca_cert.txt
```

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 Chapter 5 [Authentication], page 15. 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 4/[–tls-key-file], page 10, [tls_cert_file], page 4/[–tls-cert-file], page 10. 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 sent to the server, it probably was not issued by any CA that the server trusts.

If you need to fine tune TLS parameters or have problems connecting to your server, have a look at the [tls_force_sslv3], page 5, [tls_min_dh_prime_bits], page 5, and [tls_priorities], page 5 commands.

5 Authentication

Many SMTP servers require a client to authenticate itself before it is allowed to send mail.

Multiple authentication methods exist. Most SMTP 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 Chapter 4 [Transport Layer Security], page 13.

By default, msmtp chooses a method automatically, and it will never choose one that puts the authentication data at risk. See below for details.

msmtp supports the following authentication methods:

- '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.
- 'SCRAM-SHA-1'
  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.
- '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.
- 'GSSAPI'
  This authentication method needs a user name. The Kerberos framework takes care of secure authentication, therefore this method can safely be used without TLS.
- 'EXTERNAL'
  This is a special authentication method: The actual authentication happens outside of the SMTP protocol, typically by sending a TLS client certificate (see Chapter 4 [Transport Layer Security], page 13).
  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: 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?
- 'DIGEST-MD5'
  This is an obsolete authentication method needs a user name and a password. The authentication data is not sent in plain text, but the encryption based on MD5 is not considered secure anymore.
- '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 an obscure non-standard authentication method. It needs a user name and a password and in some cases a special domain parameter (see [ntlmdomain], page 3). The authentication data is not send in plain text, but since NTLM is not an open standard, it should be considered broken and insecure.

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 msmtp.

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, msmtp will try to find it. First, if ‘`passwordeval`’ is set, it will evaluate that command. If ‘`passwordeval`’ is not set, msmtp will try to find the password in `~/.netrc`. If that fails, it will try to find it in `SYSCONFDIR/netrc` (use `--version` to find out what `SYSCONFDIR` is on your platform). If that fails, it will try to get it from a system specific keyring (if available). If that fails but a controlling terminal is available, msmtp will prompt you for it.

Currently supported keyrings are the Gnome Keyring and the Mac OS X Keychain. The script `msmtp-gnome-tool.py` can be used to manage Gnome Keyring passwords for msmtp. To manage Mac OS X Keychain passwords, use the Keychain Access GUI application. The ‘`account name`’ is same as the msmtp ‘`user`’ argument. The ‘`keychain item name`’ is `smtp://<hostname>` where `<hostname>` matches the msmtp ‘`host`’ argument.

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 msmtp choose the method itself. See [auth], page 3, [–auth], page 9.

If msmtp chooses the method itself, it will never choose an insecure method. If TLS is active, all methods are considered secure in this context, because the connection to the server is protected by TLS. If TLS is not active, only the SCRAM-SHA-1, CRAM-MD5, and GSSAPI methods are considered secure in this context, because all the others methods put the authentication data at risk.

If you really want to risk your authentication data, you have to force msmtp to do that by manually setting the authentication method while TLS is off.

6 Delivery Status Notifications

In situations such as delivery failure or very long delivery delay, the mail system often generates a message for the sender of the mail in question, informing him about the difficulties.

Delivery Status Notification (DSN) requests, defined in RFC 3461, try to give the sender of the mail control about how and when these DSN messages are sent. The SMTP server must support the DSN extension. See Chapter 8 [Server information mode], page 20.

A first parameter controls when such messages should be generated: never, on delivery failure, on delivery delay, and/or on success. This can be set with '`dsn_notify`'/'`--dsn-notify`', see [dsn_notify], page 6/[–dsn-notify], page 10.

A second parameter controls how much of the original mail should be contained in a DSN message: only the headers, or the full mail. This can be set with '`dsn_return`'/'`--dsn-return`', see [dsn_return], page 6/[–dsn-return], page 10. Note that this parameter only applies to DSNs that indicate delivery failure for at least one recipient. If a DSN contains no indications of delivery failure, only the headers of the message are returned.

7 Sendmail mode

7.1 Envelope-from address

The SMTP server expects a sender mail address for each mail. This is the envelope-from address. It is independent of the From header (because it is part of the mail *envelope*, not of the mail itself), but in most cases both addresses are the same.

Envelope-from addresses can be generated automatically (when '`auto_from`' is enabled with the '`auto_from`' command or '`--auto-from`' option) or set explicitly with the '`from`' command and '`--from`' option. See [auto_from], page 5, [from], page 5.

When '`auto_from`' is enabled, an envelope-from address of the form user@domain will be generated. The local part will be set to `USER` or, if that fails, to `LOGNAME` or, if that fails, to the login name of the current user. The domain part can be set with the '`maildomain`' command and '`--maildomain`' option (see [maildomain], page 5). If the maildomain is empty, the envelope-from address will only consist of the user name and not have a domain part.

7.2 Logging

Logging is enabled on a per account basis. If it is enabled, msmtp will generate one log line for each mail it tries to send via the account in question.

The line will include the following information:

- Host name of the SMTP server: `host=hostname`
- Whether TLS was used: `tls=(on|off)`
- Whether authentication was used: `auth=(on|off)`
- The user name used for authentication (only if authentication is used): `user=name`
- The envelope-from address: `from=address`
- The recipient addresses: `recipients=addr1,addr2,...`
- The size of the mail as transferred to the server, in bytes (only if the delivery succeeded): `mailsize=number`
- The SMTP status code and SMTP error message (only in case of failure and only if available): `smtpstatus=number`, `smtpmsg='message'`. Multiline SMTP messages will be concatenated into one line.
- The msmtp error message (only in case of failure and only if available): `errormsg='message'`
- The msmtp exit code (from `sysexits.h`; '`EX_OK`' indicates success): `exitcode=EX_...`

If a logfile is given with the '`logfile`' command or '`--logfile`' option, this log line will be prepended with the current date and time and appended to the specified file. See [logfile], page 6, [–logfile], page 11.

If syslog logging is enabled with the '`syslog`' command or '`--syslog`' option, the log line is passed to the syslog service with the specified facility. See [syslog], page 6, [–syslog], page 11.

7.3 Bcc header

msmtp transmits mails unaltered to the SMTP server, with one exception: the Bcc header(s) will be removed before the transmission. This behavior can be changed with the '`keepbcc`' command and '`--keepbcc`' option, see [keepbcc], page 6/[–keepbcc], page 11.

8 Server information mode

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

The SMTP features that can be detected are:

- SIZE
  The maximum message size that the SMTP server accepts.
- PIPELINING
  Whether certain SMTP commands may be send in groups rather than one by one. This can speed up mail transmission if the recipient list is long. This feature is used automatically.
- STARTTLS
  See Chapter 4 [Transport Layer Security], page 13.
- AUTH
  See Chapter 5 [Authentication], page 15.
- DSN
  See Chapter 6 [Delivery Status Notifications], page 17.
- ETRN
  See Chapter 9 [Remote Message Queue Starting mode], page 21.

If TLS is activated for server information mode, the following information will be printed about the SMTP 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

9 Remote Message Queue Starting mode

Remote Message Queue Starting (RMQS) is defined in RFC 1985. It is a way for a client to request that a server start the processing of its mail queues for messages that are waiting at the server for the client machine. If any messages are at the server for the client, then the server creates a new SMTP session and sends the messages at that time.

msmtp can only send the request (using the ETRN SMTP command); a mail server on the client side should then accept the connection of the remote SMTP server to receive the mail.

RMQS requests can be sent with the '`--rmqs`' option (see [–rmqs], page 8). Destinations defined in RFC 1985 are:

- *host*
  Request the messages for the given host.
- `@`*domain*
  Request the messages for the given domain.
- *#queue*
  Request the delivery of the messages in the given queue.

10 Examples

10.1 A system wide configuration file

```
# A system wide configuration is optional.
# If it exists, it usually defines a default account.
# This allows msmtp to be used like /usr/sbin/sendmail.
account default

# The SMTP smarthost.
host mailhub.oursite.example

# Construct envelope-from addresses of the form "user@oursite.example".
#auto_from on
#maildomain oursite.example

# Use TLS.
#tls on
#tls_trust_file /etc/ssl/certs/ca-certificates.crt

# Syslog logging with facility LOG_MAIL instead of the default LOG_USER.
syslog LOG_MAIL
```

10.2 A user configuration file

```
# Set default values for all following accounts.
defaults
tls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
logfile ~/.msmtp.log

# A freemail service
account freemail
host smtp.freemail.example
from joe_smith@freemail.example
auth on
user joe.smith
password secret

# A second mail address at the same freemail service
account freemail2 : freemail
from joey@freemail.example

# The SMTP server of the provider.
account provider
host mail.provider.example
from smithjoe@provider.example
```

```
auth on
user 123
passwordeval gpg -d ~/.msmtp.password.gpg

# Set a default account
account default : provider
```

10.3 Using msmtp with Mutt

Create a configuration file for msmtp and add the following lines to your Mutt configuration file:

```
set sendmail="/path/to/msmtp"
set use_from=yes
set realname="Your Name"
set from=you@example.com
set envelope_from=yes
```

The ‘`envelope_from=yes`’ option lets Mutt use the ‘`-f`’ option of msmtp. Therefore msmtp chooses the first account that matches the from address you@example.com. Alternatively, you can use the ‘`-a`’ option:

```
set sendmail="/path/to/msmtp -a my_account"
```

Or set everything from the command line:

```
set sendmail="/path/to/msmtp --host=mailhub -f me@example.com --tls"
```

See [Choosing an account], page 11.

If you have multiple mail accounts in your msmtp configuration file and let Mutt use the ‘`-f`’ option to choose one, you can easily switch accounts in Mutt with the following Mutt configuration lines:

```
macro generic "<esc>1" ":set from=you@example.com"
macro generic "<esc>2" ":set from=you@your-employer.example"
macro generic "<esc>3" ":set from=you@some-other-provider.example"
```

Now you can use <esc>1, <esc>2, and <esc>3 to switch accounts.

The following example uses a different approach: it maps the single key `<tab>` in Compose context for switching between the various account in a handy visual way. In the same Compose context, `=` is mapped in order to show the current msmtp account. This example was contributed by Thomas Baruchel.

```
# Define <tab> and = in order to switch or see the current msmtp account
# Don't forget to put the right path for msmtp binary
macro compose \Cx_ ":set sendmail"
macro compose \Cx| "\Cx_ = \"/usr/local/bin/msmtp"
macro compose \Cx& ":macro compose \\t \\Cx"
macro compose <tab> "\Cx0"
macro compose = "\Cx_\n"
# Put the account in the following lines (here three accounts)
# Don't forget to put the number of the account at the beginning
# of the line, and the number of the next account after the '&'
macro compose \Cx0 "\Cx|\"\n\Cx&1\n\Cx_\n" # default and switch to 1
```

```
macro compose \Cx1 "\Cx| -a example_account\"\n\Cx&2\n\Cx_\n" # switch to 2
macro compose \Cx2 "\Cx| -a gmail\"\n\Cx&0\n\Cx_\n" # switch to 0
# End of the accounts
```

10.4 Using msmtp with mail

Define a default account, and put the following into `~/.mailrc`:

```
set sendmail="/path/to/msmtp"
```

You need to define a default account, because mail does not allow extra options to the msmtp command line.

10.5 Aliases file

```
# Example aliases file

# Send root to Joe and Jane
root: joe_smith@example.com, jane_chang@example.com

# Send cron to Mark
cron: mark_jones@example.com

# Send everything else to admin
default: admin@domain.example
```