



Linux Ubuntu 26.04 Manual Pages on command 'postconf.1'

\$ man postconf.1

POSTCONF(1) General Commands Manual POSTCONF(1)

NAME

postconf - Postfix configuration utility

SYNOPSIS

Managing main.cf:

postconf [-dfhHnopqvx] [-c config_dir] [-C class,...] [parameter ...]

postconf [-epv] [-c config_dir] parameter=value ...

postconf -# [-pv] [-c config_dir] parameter ...

postconf -X [-pv] [-c config_dir] parameter ...

Managing master.cf service entries:

postconf -M [-foqvx] [-c config_dir] [service[/type] ...]

postconf -M [-ev] [-c config_dir] service/type=value ...

postconf -M# [-v] [-c config_dir] service/type ...

postconf -MX [-v] [-c config_dir] service/type ...

Managing master.cf service fields:

postconf -F [-fhHoqvx] [-c config_dir] [service[/type[/field]] ...]

postconf -F [-ev] [-c config_dir] service/type/field=value ...

Managing master.cf service parameters:

postconf -P [-fhHoqvx] [-c config_dir] [service[/type[/parameter]] ...]

postconf -P [-ev] [-c config_dir] service/type/parameter=value ...

postconf -PX [-v] [-c config_dir] service/type/parameter ...

Managing bounce message templates:

postconf -b [-v] [-c config_dir] [template_file]

```
postconf -t [-v] [-c config_dir] [template_file]
```

Managing TLS features:

```
postconf -T mode [-v] [-c config_dir]
```

Managing other configuration:

```
postconf -a|-A|-l|-m [-v] [-c config_dir]
```

DESCRIPTION

By default, the `postconf(1)` command displays the values of `main.cf` configuration parameters, and warns about possible mis-typed parameter names (Postfix 2.9 and later). The command can also change `main.cf` configuration parameter values, or display other configuration information about the Postfix mail system.

Options:

-a List the available SASL plug-in types for the Postfix SMTP server. The plug-in type is selected with the `smtpd_sasl_type` configuration parameter by specifying one of the names listed below.

`cyrus` This server plug-in is available when Postfix is built with Cyrus SASL support.
`port`.

`dovecot`

This server plug-in uses the Dovecot authentication server, and is available when Postfix is built with any form of SASL support.

This feature is available with Postfix 2.3 and later.

-A List the available SASL plug-in types for the Postfix SMTP client. The plug-in type is selected with the `smtp_sasl_type` or `lmtp_sasl_type` configuration parameters by specifying one of the names listed below.

`cyrus` This client plug-in is available when Postfix is built with Cyrus SASL support.
`port`.

This feature is available with Postfix 2.3 and later.

-b [template_file]

Display the message text that appears at the beginning of delivery status notification (DSN) messages, expanding `$name` expressions with actual values as described in `bounce(5)`.

To override the `bounce_template_file` parameter setting, specify a template file name at the end of the "`postconf -b`" command line. Specify an empty file name to display built-in templates (in shell language: `""`).

This feature is available with Postfix 2.3 and later.

-c config_dir

The main.cf configuration file is in the named directory instead of the default configuration directory.

-C class,...

When displaying main.cf parameters, select only parameters from the specified class(es):

builtin

Parameters with built-in names.

service

Parameters with service-defined names (the first field of a master.cf entry plus a Postfix-defined suffix).

user Parameters with user-defined names.

all All the above classes.

The default is as if "-C all" is specified.

This feature is available with Postfix 2.9 and later.

-d Print main.cf default parameter settings instead of actual settings. Specify -df to fold long lines for human readability (Postfix 2.9 and later).

-e Edit the main.cf configuration file, and update parameter settings with the "name=value" pairs on the postconf(1) command line.

With -M, edit the master.cf configuration file, and replace one or more service entries with new values as specified with "service/type=value" on the postconf(1) command line.

With -F, edit the master.cf configuration file, and replace one or more service fields with new values as specified with "service/type/field=value" on the postconf(1) command line. Currently, the "command" field contains the command name and command arguments. This may change in the near future, so that the "command" field contains only the command name, and a new "arguments" pseudofield contains the command arguments.

With -P, edit the master.cf configuration file, and add or update one or more service parameter settings (-o parameter=value settings) with new values as specified with "service/type/parameter=value" on the postconf(1) command line.

In all cases the file is copied to a temporary file then renamed into place. Specify

quotes to protect special characters and whitespace on the `postconf(1)` command line.

The `-e` option is no longer needed with Postfix version 2.8 and later, as it is assumed whenever a value is specified (empty or non-empty).

- f Fold long lines when printing `main.cf` or `master.cf` configuration file entries, for human readability.

This feature is available with Postfix 2.9 and later.

- F Show `master.cf` per-entry field settings (by default all services and all fields), formatted as "service/type/field=value", one per line. Specify `-Ff` to fold long lines.

Specify one or more "service/type/field" instances on the `postconf(1)` command line to limit the output to fields of interest. Trailing parameter name or service type fields that are omitted will be handled as "*" wildcard fields.

This feature is available with Postfix 2.11 and later.

- h Show parameter or attribute values without the "name = " label that normally precedes the value.
- H Show parameter or attribute names without the " = value" that normally follows the name.

This feature is available with Postfix 3.1 and later.

- l List the names of all supported mailbox locking methods. Postfix supports the following methods:

`flock` A kernel-based advisory locking method for local files only. This locking method is available on systems with a BSD compatible library.

`fcntl` A kernel-based advisory locking method for local and remote files.

`dotlock`

An application-level locking method. An application locks a file named `filename` by creating a file named `filename.lock`. The application is expected to remove its own lock file, as well as stale lock files that were left behind after abnormal program termination.

- m List the names of all supported lookup table types. In Postfix configuration files, lookup tables are specified as `type:name`, where `type` is one of the types listed below. The table name syntax depends on the lookup table type as described in the `DATA?BASE_README` document.

`btree` A sorted, balanced tree structure. Available on systems with support for

Berkeley DB databases.

`cdb` A read-optimized structure with no support for incremental updates. Available on systems with support for CDB databases.

This feature is available with Postfix 2.2 and later.

`cidr` A table that associates values with Classless Inter-Domain Routing (CIDR) patterns. This is described in `cidr_table(5)`.

This feature is available with Postfix 2.2 and later.

`dbm` An indexed file type based on hashing. Available on systems with support for DBM databases.

`environ`

The UNIX process environment array. The lookup key is the environment variable name; the table name is ignored. Originally implemented for testing, someone may find this useful someday.

`fail` A table that reliably fails all requests. The lookup table name is used for logging. This table exists to simplify Postfix error tests.

This feature is available with Postfix 2.9 and later.

`hash` An indexed file type based on hashing. Available on systems with support for Berkeley DB databases.

`inline (read-only)`

A non-shared, in-memory lookup table. Example: `"inline:{ key=value, { key = text with whitespace or comma } }"`. Key-value pairs are separated by whitespace or comma; with a key-value pair inside `"{"`, whitespace is ignored after the opening `"{"`, around the `"="` between key and value, and before the closing `"}"`.

Inline tables eliminate the need to create a database file for just a few fixed elements. See also the `static: map` type.

This feature is available with Postfix 3.0 and later.

`internal`

A non-shared, in-memory hash table. Its content are lost when a process terminates.

`lmdb` OpenLDAP LMDB database (a memory-mapped, persistent file). Available on systems with support for LMDB databases. This is described in `lmdb_table(5)`.

This feature is available with Postfix 2.11 and later.

`ldap (read-only)`

LDAP database client. This is described in `ldap_table(5)`.

memcache

Memcache database client. This is described in `memcache_table(5)`.

This feature is available with Postfix 2.9 and later.

mongodb

MongoDB database client. This is described in `mongodb_table(5)`.

This feature is available with Postfix 3.9 and later.

mysql (read-only)

MySQL database client. Available on systems with support for MySQL databases.

This is described in `mysql_table(5)`.

pcre (read-only)

A lookup table based on Perl Compatible Regular Expressions. The file format is described in `pcre_table(5)`.

pgsql (read-only)

PostgreSQL database client. This is described in `pgsql_table(5)`.

This feature is available with Postfix 2.1 and later.

pipemap (read-only)

A lookup table that constructs a pipeline of tables. Example:

"pipemap:{type_1:name_1, ..., type_n:name_n}". Each "pipemap:" query is given to the first table. Each lookup result becomes the query for the next table in the pipeline, and the last table produces the final result. When any table lookup produces no result, the pipeline produces no result. The first and last characters of the "pipemap:" table name must be "{" and "}". Within these, individual maps are separated with comma or whitespace.

This feature is available with Postfix 3.0 and later.

proxy Postfix `proxymap(8)` client for shared access to Postfix databases. The table name syntax is `type:name`.

This feature is available with Postfix 2.0 and later.

randmap (read-only)

An in-memory table that performs random selection. Example: "randmap:{result_1, ..., result_n}". Each table query returns a random choice from the specified results. The first and last characters of the "randmap:" table name must be "{" and "}". Within these, individual results are separated with

comma or whitespace. To give a specific result more weight, specify it multiple times.

This feature is available with Postfix 3.0 and later.

regexp (read-only)

A lookup table based on regular expressions. The file format is described in `regexp_table(5)`.

sdbm An indexed file type based on hashing. Available on systems with support for SDBM databases.

This feature is available with Postfix 2.2 and later.

socketmap (read-only)

Sendmail-style socketmap client. The table name is `inet:host:port:name` for a TCP/IP server, or `unix:pathname:name` for a UNIX-domain server. This is described in `socketmap_table(5)`.

This feature is available with Postfix 2.10 and later.

sqlite (read-only)

SQLite database. This is described in `sqlite_table(5)`.

This feature is available with Postfix 2.8 and later.

static (read-only)

A table that always returns its name as lookup result. For example, `static:foobar` always returns the string `foobar` as lookup result. Specify `"static:{ text with whitespace }"` when the result contains whitespace; this form ignores whitespace after the opening `"{"` and before the closing `"}"`. See also the inline: `map`.

The form `"static:{text}"` is available with Postfix 3.0 and later.

tcp (read-only)

TCP/IP client. The protocol is described in `tcp_table(5)`.

texthash (read-only)

Produces similar results as `hash: files`, except that you don't need to run the `postmap(1)` command before you can use the file, and that it does not detect changes after the file is read.

This feature is available with Postfix 2.8 and later.

unionmap (read-only)

A table that sends each query to multiple lookup tables and that concatenates

all found results, separated by comma. The table name syntax is the same as for pipemap.

This feature is available with Postfix 3.0 and later.

unix (read-only)

A limited view of the UNIX authentication database. The following tables are implemented:

unix:passwd.byname

The table is the UNIX password database. The key is a login name. The result is a password file entry in passwd(5) format.

unix:group.byname

The table is the UNIX group database. The key is a group name. The result is a group file entry in group(5) format.

Other table types may exist depending on how Postfix was built.

-M Show master.cf file contents instead of main.cf file contents. Specify **-Mf** to fold long lines for human readability.

Specify zero or more arguments, each with a service-name or service-name/service-type pair, where service-name is the first field of a master.cf entry and service-type is one of (inet, unix, fifo, or pass).

If service-name or service-name/service-type is specified, only the matching master.cf entries will be output. For example, "postconf -Mf smtp" will output all services named "smtp", and "postconf -Mf smtp/inet" will output only the smtp service that listens on the network. Trailing service type fields that are omitted will be handled as "*" wildcard fields.

This feature is available with Postfix 2.9 and later. The syntax was changed from "name.type" to "name/type", and "*" wildcard support was added with Postfix 2.11.

-n Show only configuration parameters that have explicit name=value settings in main.cf. Specify **-nf** to fold long lines for human readability (Postfix 2.9 and later). To show settings that differ from built-in defaults only, use the following bash syntax:

```
LANG=C comm -23 <(postconf -n) <(postconf -d)
```

Replace "-23" with "-12" to show settings that duplicate built-in defaults.

-o name=value

Override main.cf parameter settings. This lets you see the effect changing a parameter would have when it is used in other configuration parameters, e.g.:

postconf -x -o stress=yes

This feature is available with Postfix 2.10 and later.

-p Show main.cf parameter settings. This is the default.

This feature is available with Postfix 2.11 and later.

-P Show master.cf service parameter settings (by default all services and all parameters), formatted as "service/type/parameter=value", one per line. Specify -Pf to fold long lines.

Specify one or more "service/type/parameter" instances on the postconf(1) command line to limit the output to parameters of interest. Trailing parameter name or service type fields that are omitted will be handled as "*" wildcard fields.

This feature is available with Postfix 2.11 and later.

-q Do not log warnings for deprecated or unused parameters.

This feature is available with Postfix 3.9 and later.

-t [template_file]

Display the templates for text that appears at the beginning of delivery status notification (DSN) messages, without expanding \$name expressions.

To override the bounce_template_file parameter setting, specify a template file name at the end of the "postconf -t" command line. Specify an empty file name to display built-in templates (in shell language: "").

This feature is available with Postfix 2.3 and later.

-T mode

If Postfix is compiled without TLS support, the -T option produces no output. Otherwise, if an invalid mode is specified, the -T option reports an error and exits with a non-zero status code. The valid modes are:

compile-version

Output the OpenSSL version that Postfix was compiled with (i.e. the OpenSSL version in a header file). The output format is the same as with the command "openssl version".

run-version

Output the OpenSSL version that Postfix is linked with at runtime (i.e. the OpenSSL version in a shared library).

public-key-algorithms

Output the lower-case names of the supported public-key algorithms, one

per-line.

This feature is available with Postfix 3.1 and later.

-v Enable verbose logging for debugging purposes. Multiple -v options make the software increasingly verbose.

-x Expand \$name in main.cf or master.cf parameter values. The expansion is recursive.

This feature is available with Postfix 2.10 and later.

-X Edit the main.cf configuration file, and remove the parameters named on the postconf(1) command line. Specify a list of parameter names, not "name=value" pairs.

With -M, edit the master.cf configuration file, and remove one or more service entries as specified with "service/type" on the postconf(1) command line.

With -P, edit the master.cf configuration file, and remove one or more service parameter settings (-o parameter=value settings) as specified with "service/type/parameter" on the postconf(1) command line.

In all cases the file is copied to a temporary file then renamed into place. Specify quotes to protect special characters on the postconf(1) command line.

There is no postconf(1) command to perform the reverse operation.

This feature is available with Postfix 2.10 and later. Support for -M and -P was added with Postfix 2.11.

-# Edit the main.cf configuration file, and comment out the parameters named on the postconf(1) command line, so that those parameters revert to their default values. Specify a list of parameter names, not "name=value" pairs.

With -M, edit the master.cf configuration file, and comment out one or more service entries as specified with "service/type" on the postconf(1) command line.

In all cases the file is copied to a temporary file then renamed into place. Specify quotes to protect special characters on the postconf(1) command line.

There is no postconf(1) command to perform the reverse operation.

This feature is available with Postfix 2.6 and later. Support for -M was added with Postfix 2.11.

DIAGNOSTICS

Problems are reported to the standard error stream.

ENVIRONMENT

MAIL_CONFIG

Directory with Postfix configuration files.

CONFIGURATION PARAMETERS

The following main.cf parameters are especially relevant to this program.

The text below provides only a parameter summary. See `postconf(5)` for more details including examples.

`config_directory` (see '`postconf -d`' output)

The default location of the Postfix main.cf and master.cf configuration files.

`bounce_template_file` (empty)

Pathname of a configuration file with bounce message templates.

FILES

`/etc/postfix/main.cf`, Postfix configuration parameters

`/etc/postfix/master.cf`, Postfix master daemon configuration

SEE ALSO

`bounce(5)`, bounce template file format

`master(5)`, master.cf configuration file syntax

`postconf(5)`, main.cf configuration file syntax

README FILES

Use "`postconf readme_directory`" or "`postconf html_directory`" to locate this information.

`DATABASE_README`, Postfix lookup table overview

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema

IBM T.J. Watson Research

P.O. Box 704

Yorktown Heights, NY 10598, USA

Wietse Venema

Google, Inc.

111 8th Avenue

New York, NY 10011, USA

POSTCONF(1)