NAME

dirmngr − CRL and OCSP daemon

SYNOPSIS

dirmngr [*options*] *command* [*args*]

DESCRIPTION

Since version 2.1 of GnuPG, **dirmngr** takes care of accessing the OpenPGP keyservers. As with previous versions it is also used as a server for managing and downloading certificate revocation lists (CRLs) for X.509 certificates, downloading X.509 certificates, and providing access to OCSP providers. Dirmngr is invoked internally by **gpg**, **gpgsm**, or via the **gpg-connect-agent** tool.

COMMANDS

Commands are not distinguished from options except for the fact that only one command is allowed.

--version

Print the program version and licensing information. Note that you cannot abbreviate this command.

--help, -h

Print a usage message summarizing the most useful command-line options. Note that you cannot abbreviate this command.

--dump-options

Print a list of all available options and commands. Note that you cannot abbreviate this command.

--server

Run in server mode and wait for commands on the **stdin**. The default mode is to create a socket and listen for commands there. This is only used for testing.

--daemon

Run in background daemon mode and listen for commands on a socket. This is the way **dirmngr** is started on demand by the other GnuPG components. To force starting **dirmngr** it is in general best to use **gpgconf --launch dirmngr**.

--supervised

Run in the foreground, sending logs to stderr, and listening on file descriptor 3, which must already be bound to a listening socket. This is useful when running under systemd or other similar process supervision schemes. This option is not supported on Windows.

--list-crls

List the contents of the CRL cache on **stdout**. This is probably only useful for debugging purposes.

--load-crl *file*

This command requires a filename as additional argument, and it will make Dirmngr try to import the CRL in *file* into it's cache. Note, that this is only possible if Dirmngr is able to retrieve the

CA's certificate directly by its own means. In general it is better to use **gpgsm**'s **--call-dirmngr loadcrl filename** command so that **gpgsm** can help dirmngr.

--fetch-crl *url*
This command requires an URL as additional argument, and it will make dirmngr try to retrieve and import the CRL from that *url* into it's cache. This is mainly useful for debugging purposes. The **dirmngr-client** provides the same feature for a running dirmngr.

--shutdown
This commands shuts down an running instance of Dirmngr. This command has currently no effect.

--flush This command removes all CRLs from Dirmngr's cache. Client requests will thus trigger reading of fresh CRLs.

OPTIONS

Note that all long options with the exception of **--options** and **--homedir** may also be given in the configuration file after stripping off the two leading dashes.

--options *file*
Reads configuration from *file* instead of from the default per-user configuration file. The default configuration file is named '*dirmngr.conf*' and expected in the home directory.

--homedir *dir*
Set the name of the home directory to *dir*. This option is only effective when used on the command line. The default is the directory named '*.gnupg*' directly below the home directory of the user unless the environment variable **GNUPGHOME** has been set in which case its value will be used. Many kinds of data are stored within this directory.

-v

--verbose
Outputs additional information while running. You can increase the verbosity by giving several verbose commands to **dirmngr**, such as **-vv**.

--log-file *file*
Append all logging output to *file*. This is very helpful in seeing what the agent actually does. Use '*socket://*' to log to socket.

--debug-level *level*
Select the debug level for investigating problems. *level* may be a numeric value or by a keyword:

none No debugging at all. A value of less than 1 may be used instead of the keyword.

basic Some basic debug messages. A value between 1 and 2 may be used instead of the keyword.

advanced
More verbose debug messages. A value between 3 and 5 may be used instead of the keyword.

expert Even more detailed messages. A value between 6 and 8 may be used instead of the keyword.

guru All of the debug messages you can get. A value greater than 8 may be used instead of the keyword. The creation of hash tracing files is only enabled if the keyword is used.

How these messages are mapped to the actual debugging flags is not specified and may change with newer releases of this program. They are however carefully selected to best aid in debugging.

--debug *flags*
Set debugging flags. This option is only useful for debugging and its behavior may change with a new release. All flags are or-ed and may be given in C syntax (e.g. 0x0042) or as a comma separated list of flag names. To get a list of all supported flags the single word "help" can be used.

--debug-all
Same as **--debug=0xffffffff**

--tls-debug *level*
Enable debugging of the TLS layer at *level*. The details of the debug level depend on the used TLS library and are not set in stone.

--debug-wait *n*
When running in server mode, wait *n* seconds before entering the actual processing loop and print the pid. This gives time to attach a debugger.

--disable-check-own-socket
On some platforms **dirmngr** is able to detect the removal of its socket file and shutdown itself. This option disable this self-test for debugging purposes.

-s
--sh
-c
--csh Format the info output in daemon mode for use with the standard Bourne shell respective the C-shell. The default is to guess it based on the environment variable **SHELL** which is in almost all cases sufficient.

--force Enabling this option forces loading of expired CRLs; this is only useful for debugging.

--use-tor
--no-use-tor
The option **--use-tor** switches Dirmngr and thus GnuPG into ‘‘Tor mode’’ to route all network access via Tor (an anonymity network). Certain other features are disabled in this mode. The effect of **--use-tor** cannot be overridden by any other command or even by reloading dirmngr. The use of **--no-use-tor** disables the use of Tor. The default is to use Tor if it is available on startup or after reloading dirmngr.

--standard-resolver
This option forces the use of the system's standard DNS resolver code. This is mainly used for debugging. Note that on Windows a standard resolver is not used and all DNS access will return the error ''Not Implemented'' if this option is used. Using this together with enabled Tor mode returns the error ''Not Enabled''.

--recursive-resolver
When possible use a recursive resolver instead of a stub resolver.

--resolver-timeout *n*
Set the timeout for the DNS resolver to N seconds. The default are 30 seconds.

--connect-timeout *n*

--connect-quick-timeout *n*
Set the timeout for HTTP and generic TCP connection attempts to N seconds. The value set with the quick variant is used when the --quick option has been given to certain Assuan commands. The quick value is capped at the value of the regular connect timeout. The default values are 15 and 2 seconds. Note that the timeout values are for each connection attempt; the connection code will attempt to connect all addresses listed for a server.

--listen-backlog *n*
Set the size of the queue for pending connections. The default is 64.

--allow-version-check
Allow Dirmngr to connect to **https://versions.gnupg.org** to get the list of current software versions. On debian-packaged versions, this option does nothing since software updates should be handled by the distribution. See the option**--query-swdb** of the command **gpgconf** for more details. Note, that regardless of this option a version check can always be triggered using this command:

```
gpg-connect-agent --dirmngr 'loadswdb --force' /bye
```

--keyserver *name*
Use *name* as your keyserver. This is the server that **gpg** communicates with to receive keys, send keys, and search for keys. The format of the*name* is a URI: 'scheme:[//]k eyservername[:port]' The scheme is the type of keyserver: "hkp" for the HTTP (or compatible) keyservers, "ldap" for the LDAP keyservers, or "mailto" for the Graff email keyserver. Note that your particular installation of GnuPG may have other keyserver types available as well. Keyserver schemes are case-insensitive. After the keyserver name, optional keyserver configuration options may be provided. These are the same as the **--keyserver-options** of **gpg**, but apply only to this particular keyserver.

Most keyservers synchronize with each other, so there is generally no need to send keys to more than one server. The keyserver **hkp://keys.gnupg.net** uses round robin DNS to give a different keyserver each time you use it.

If exactly two keyservers are configured and only one is a Tor hidden service (.onion), Dirmngr selects the keyserver to use depending on whether Tor is locally running or not. The check for a running Tor is done for each new connection.

If no keyserver is explicitly configured, dirmngr will use the built-in default of **hkps://keys.openpgp.org**.

Note that the above default is a Debian-specific choice. Upstream GnuPG prefers **hkps://hkps.pool.sks-keyservers.net**. See /usr/share/doc/gpgconf/NEWS.Debian.gz for more details.

Windows users with a keyserver running on their Active Directory should use **ldap:///** for *name* to access this directory.

For accessing anonymous LDAP keyservers *name* is in general just a **ldaps://ldap.example.com**. A BaseDN parameter should never be specified. If authentication is required the value of *name* is for example:

keyserver ldaps://ldap.example.com/????bindname=uid=USERNAME
%2Cou=GnuPG%20Users%2Cdc=example%2Cdc=com,password=PASSWORD

Put this all on one line without any spaces and keep the '%2C' as given. Replace USERNAME, PASSWORD, and the 'dc' parts according to the instructions received from the LDAP administrator. Note that only simple authentication (i.e. cleartext passwords) is supported and thus using ldaps is strongly suggested.

--nameserver *ipaddr*
In ''Tor mode'' Dirmngr uses a public resolver via Tor to resolve DNS names. If the default public resolver, which is **8.8.8.8**, shall not be used a different one can be given using this option. Note that a numerical IP address must be given (IPv6 or IPv4) and that no error checking is done for *ipaddr*.

--disable-ipv4

--disable-ipv6
Disable the use of all IPv4 or IPv6 addresses.

--disable-ldap
Entirely disables the use of LDAP.

--disable-http
Entirely disables the use of HTTP.

--ignore-http-dp
When looking for the location of a CRL, the to be tested certificate usually contains so called CRL Distribution Point (DP) entries which are URLs describing the way to access the CRL. The first found DP entry is used. With this option all entries using the HTTP scheme are ignored when looking for a suitable DP.

--ignore-ldap-dp
This is similar to **--ignore-http-dp** but ignores entries using the LDAP scheme. Both options may be combined resulting in ignoring DPs entirely.

--ignore-ocsp-service-url
Ignore all OCSP URLs contained in the certificate. The effect is to force the use of the default responder.

--honor-http-proxy
If the environment variable '*http_proxy*' has been set, use its value to access HTTP servers.

--http-proxy *host*[:*port*]
Use *host* and *port* to access HTTP servers. The use of this option overrides the environment variable '*http_proxy*' regardless whether **--honor-http-proxy** has been set.

--ldap-proxy *host*[:*port*]
Use *host* and *port* to connect to LDAP servers. If*port* is omitted, port 389 (standard LD AP port) is used. This overrides any specified host and port part in a LDAP URL and will also be used if host and port have been omitted from the URL.

--only-ldap-proxy
Never use anything else but the LDAP "proxy" as configured with **--ldap-proxy**. Usually**dirmngr** tries to use other configured LDAP server if the connection using the "proxy" failed.

--ldapserverlist-file *file*
Read the list of LDAP servers to consult for CRLs and certificates from file instead of the default per-user ldap server list file. The default value for *file* is '*dirmngr_ldapservers.conf*'.

This server list file contains one LDAP server per line in the format

hostname:port:username:password:base_dn

Lines starting with a '#' are comments.

Note that as usual all strings entered are expected to be UTF-8 encoded. Obviously this will lead to problems if the password has originally been encoded as Latin-1. There is no other solution here than to put such a password in the binary encoding into the file (i.e. non-ascii characters won't show up readable). ([The **gpgconf** tool might be helpful for frontends as it enables editing this configuration file using percent-escaped strings.])

--ldaptimeout *secs*
Specify the number of seconds to wait for an LDAP query before timing out. The default are 15 seconds. 0 will never timeout.

--add-servers
This option makes dirmngr add any servers it discovers when validating certificates against CRLs to the internal list of servers to consult for certificates and CRLs.

This option is useful when trying to validate a certificate that has a CRL distribution point that points to a server that is not already listed in the ldapserverlist. Dirmngr will always go to this server and try to download the CRL, but chances are high that the certificate used to sign the CRL is located on the same server. So if dirmngr doesn't add that new server to list, it will often not be able to verify the signature of the CRL unless the **--add-servers** option is used.

Note: The current version of dirmngr has this option disabled by default.

--allow-ocsp
This option enables OCSP support if requested by the client.

OCSP requests are rejected by default because they may violate the privacy of the user; for example it is possible to track the time when a user is reading a mail.

--ocsp-responder *url*
Use *url* as the default OCSP Responder if the certificate does not contain information about an assigned responder. Note, that**--ocsp-signer** must also be set to a v alid certificate.

--ocsp-signer *fpr|file*
Use the certificate with the fingerprint *fpr* to check the responses of the default OCSP Responder. Alternatively a filename can be given in which case the response is expected to be signed by one of the certificates described in that file. Any argument which contains a slash, dot or tilde is considered a filename. Usual filename expansion takes place: A tilde at the start followed by a slash is replaced by the content of '*HOME*', no slash at start describes a relative filename which will be searched at the home directory. To make sure that the *file* is searched in the home directory, either prepend the name with "./" or use a name which contains a dot.

If a response has been signed by a certificate described by these fingerprints no further check upon the validity of this certificate is done.

The format of the *FILE* is a list of SHA-1 fingerprint, one per line with optional colons between the bytes. Empty lines and lines prefix with a hash mark are ignored.

--ocsp-max-clock-skew *n*
The number of seconds a skew between the OCSP responder and them local clock is accepted. Default is 600 (10 minutes).

--ocsp-max-period *n*
Seconds a response is at maximum considered valid after the time given in the thisUpdate field. Default is 7776000 (90 days).

--ocsp-current-period *n*
The number of seconds an OCSP response is considered valid after the time given in the NEXT_UPDATE datum. Default is 10800 (3 hours).

--max-replies *n*
Do not return more that *n* items in one query. The default is 10.

--ignore-cert-extension *oid*
Add *oid* to the list of ignored certificate extensions. The*oid* is e xpected to be in dotted decimal form, like **2.5.29.3**. This option may be used more than once. Critical flagged certificate extensions matching one of the OIDs in the list are treated as if they are actually handled and thus the certificate won't be rejected due to an unknown critical extension. Use this option with care because extensions are usually flagged as critical for a reason.

--hkp-cacert *file*
Use the root certificates in *file* for verification of the TLS certificates used with **hkps** (keyserver access over TLS). If the file is in PEM format a suffix of **.pem** is expected for *file*. This option may be given multiple times to add more root certificates. Tilde expansion is supported.

If no **hkp-cacert** directive is present, dirmngr will make a reasonable choice: if the keyserver in question is the special pool **hkps.pool.sks-keyservers.net**, it will use the bundled root certificate for that pool. Otherwise, it will use the system CAs.

EXAMPLES

Here is an example on how to show dirmngr's internal table of OpenPGP keyserver addresses. The output is intended for debugging purposes and not part of a defined API.

```
gpg-connect-agent --dirmngr 'keyserver --hosttable' /bye
```

To inhibit the use of a particular host you have noticed in one of the keyserver pools, you may use

```
gpg-connect-agent --dirmngr 'keyserver --dead pgpkeys.bnd.de' /bye
```

The description of the **keyserver** command can be printed using

```
gpg-connect-agent --dirmngr 'help keyserver' /bye
```

FILES

Dirmngr makes use of several directories when running in daemon mode: There are a few configuration files whih control the operation of dirmngr. By default they may all be found in the current home directory (see: [option --homedir]).

dirmngr.conf
This is the standard configuration file read by **dirmngr** on startup. It may contain any valid long option; the leading two dashes may not be entered and the option may not be abbreviated. This file is also read after a **SIGHUP** however not all options will actually have an effect. This default name may be changed on the command line (see: [option --options]). You should backup this file.

/etc/gnupg/trusted-certs
This directory should be filled with certificates of Root CAs you are trusting in checking the CRLs and signing OCSP Responses.

Usually these are the same certificates you use with the applications making use of dirmngr. It is expected that each of these certificate files contain exactly one DER encoded certificate in a file with the suffix '*.crt*' or '*.der*'. **dirmngr** reads those certificates on startup and when given a SIGHUP. Certificates which are not readable or do not make up a proper X.509 certificate are ignored; see the log file for details.

Applications using dirmngr (e.g. gpgsm) can request these certificates to complete a trust chain in the same way as with the extra-certs directory (see below).

Note that for OCSP responses the certificate specified using the option **--ocsp-signer** is always considered valid to sign OCSP requests.

/etc/gnupg/extra-certs
This directory may contain extra certificates which are preloaded into the internal cache on startup. Applications using dirmngr (e.g. gpgsm) can request cached certificates to complete a trust chain. This is convenient in cases you have a couple intermediate CA certificates or certificates usually used to sign OCSP responses. These certificates are first tried before going out to the net to look for them. These certificates must also be DER encoded and suffixed with '*.crt*' or '*.der*'.

~/.gnupg/crls.d
This directory is used to store cached CRLs. The '*crls.d*' part will be created by dirmngr if it does not exists but you need to make sure that the upper directory exists.

SIGNALS

A running **dirmngr** may be controlled by signals, i.e. using the **kill** command to send a signal to the process.

Here is a list of supported signals:

SIGHUP
This signal flushes all internally cached CRLs as well as any cached certificates. Then the certificate cache is reinitialized as on startup. Options are re-read from the configuration file. Instead of sending this signal it is better to use
gpgconf --reload dirmngr

SIGTERM
Shuts down the process but waits until all current requests are fulfilled. If the process has received 3 of these signals and requests are still pending, a shutdown is forced. You may also use
gpgconf --kill dirmngr
instead of this signal

SIGINT
Shuts down the process immediately.

SIGUSR1
This prints some caching statistics to the log file.

SEE ALSO

gpgsm(1), **dirmngr-client**(1)

The full documentation for this tool is maintained as a Texinfo manual. If GnuPG and the info program are properly installed at your site, the command

info gnupg

should give you access to the complete manual including a menu structure and an index.