Using Dirmngr

for version 1.1.0rc1, 9 March 2010

Steffen Hansen (steffen@klaralvdalens-datakonsult.se)
Werner Koch (wk@g10code.com)

This manual is for Dirmngr (version 1.1.0rc1, 9 March 2010), which is an X.509 CRL and OCSP manager.

Table of Contents

1 About Dirmngr.

Dirmngr is a server for managing and downloading certificate revocation lists (CRLs) for X.509 certificates and for downloading the certificates themselves. Dirmngr also handles OCSP requests as an alternative to CRLs. Dirmngr is either invoked internally by gpgsm (from GnuPG 2) or when running as a system daemon through the `dirmngr-client` tool.

2 How to install Dirmngr.

Installation is decribed in the file 'INSTALL' and given that you are already reading this documentation we can only give some hints on further configuration. If you plan to use dirmngr as a system daemon and not only as a part of GnuPG 2, you should read on.

If dirmngr is started in system daemon mode, it uses a directory layout as common for system daemon and does not make use of the drefault '~/.gnupg' directory. To comply with the rules on GNU/Linux systems you should have build time configured dirmngr using:

```
./configure --sysconfdir=/etc --localstatedir=/var
```

This is to make sure that the configuration file is searched in the directory '/etc/dirmngr' and the variable data below '/var'; the default would be to install them in the '/usr/local' too where the binaries get installed. If you selected to use the '--prefix=/' you obviously don't need those option as they are the default then. Further on we assume that you used these options.

Dirmngr makes use of several directories when running in daemon mode:

'/etc/dirmngr'
: This is where all the configuration files are expected by default.

'/etc/dirmngr/trusted-certs'
: This directory should be filled with certificates of Root CAs you are trusting in checking the CRLS and signing OCSP Reponses. 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.

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

'/var/lib/dirmngr/extra-certs'
: This directory may contain extra certificates which are preloaded into the interal cache on startup. This is convenient in cases you have a couple intermediate CA certificates or certificates ususally used to sign OCSP reponses. 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'.

'/var/run/dirmngr'
: This directory keeps the socket file for accsing dirmngr services. The name of the socket file will be 'socket'. Make sure that this directory has the proper permissions to let dirmngr create the socket file and that eligible users may read and write to that socket.

'/var/cache/dirmngr/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.

To be able to see what's going on you should create the configure file '/etc/dirmngr/dirmngr.conf' with at least one line:

```
log-file /var/log/dirmngr/dirmngr.log
```

To be able to perform OCSP requests you probably want to add the line:

```
allow-ocsp
```

Now you may start dirmngr as a system daemon using:

```
dirmngr --daemon
```

Please ignore the output; it is not needed anymore. Check the log file to see whether all trusted root certificates have benn loaded correctly.

3 Commands

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

`--version`
: Print the program version and licensing information. Note that you can abbreviate this command.

`--help, -h`
: Print a usage message summarizing the most useful command-line options. Not that you can 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.

`--daemon` Run in background daemon mode and listen for commands on a socket. Note that this also changes the default home directory and enables the internal certificate validation code.

`--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 an 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 corrently no effect.

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

4 Option Summary

`--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 depends on the running mode:

  With `--daemon` given on the commandline
  : the directory named '`/etc/dirmngr`' for configuration files, '`/var/lib/dirmngr/`' for extra data and '`/var/cache/dirmngr`' for cached CRLs.

  Without `--daemon` given on the commandline
  : 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. All kind of data is stored below 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.

`--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*
: This option is only useful for debugging and the behaviour may change at any time without notice. FLAGS are bit encoded and may be given in usual C-Syntax.

`--debug-all`
: Same as `--debug=0xffffffff`

`--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.

`-s`
`--sh`
`-c`
`--csh`
: Format the info output in daemon mode for use with the standard Bourne shell respective the C-shell . The default ist 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.

`--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 options 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 ommitted, port 389 (standard LDAP 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 ommitted 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`' or '`ldapservers.conf`' when running in '`--daemon`' mode.

  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 orginally 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).[1]

`--ldaptimeout` *secs*
: Specify the number of seconds to wait for an LDAP query before timing out. The default is currently 100 seconds. 0 will never timeout.

`--add-servers`
: This options 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 options 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.

---

[1] The `gpgconf` tool might be helpful for frontends as it allows to edit this configuration file using percent escaped strings.

`--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 valid certificate.

`--ocsp-signer` *fpr|file*
: Use the certificate with the fingerprint *fpr* to check the responses of the default OCSP Responder. Alternativly a filename can be given in which case the respinse 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 (20 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 expected 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.

5 Use of 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 signals 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.

`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.

`SIGINT` Shuts down the process immediately.

`SIGUSR1` This prints some caching statistics to the log file.

6 Examples

The way to start the dirmngr in the foreground (as done by tools if no dirmngr is running in the background) is to use:

```
dirmngr --server -v
```

If a dirmngr is supposed to be used as a system wide daemon, it should be started like:

```
dirmngr --daemon
```

This will force it to go into the backround, read the default certificates (including the trusted root certificates) and listen on a socket for client requests. It does also print information about the socket used but they are only for compatibilty reasons with old GnuPG versions and may be ignored.

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

7 Dirmngr's Assuan Protocol

Assuan is the IPC protocol used to access dirmngr. This is a description of the commands implemented by dirmngr.

7.1 Return the certificate(s) found

Lookup certificate. To allow multiple patterns (which are ORed) quoting is required: Spaces are to be translated into "+" or into "%20"; obviously this requires that the usual escape quoting rules are applied. The server responds with:

```
S: D <DER encoded certificate>
S: END
S: D <second DER encoded certificate>
S: END
S: OK
```

In this example 2 certificates are returned. The server may return any number of certificates; OK will also be returned when no certificates were found. The dirmngr might return a status line

```
S: S TRUNCATED <n>
```

To indicate that the output was truncated to N items due to a limitation of the server or by an arbitrary set limit.

The option '`--url`' may be used if instead of a search pattern a complete URL to the certificate is known:

```
C: LOOKUP --url CN%3DWerner%20Koch,o%3DIntevation%20GmbH,c%3DDE?userCertificate
```

If the option '`--cache-only`' is given, no external lookup is done so that only certificates from the cache are returned.

With the option '`--single`', the first and only the first match will be returned. Unless option '`--cache-only`' is also used, no local lookup will be done in this case.

7.2 Validate a certificate using a CRL or OCSP

`ISVALID [--only-ocsp] [--force-default-responder]` *`certid|certfpr`*

Check whether the certificate described by the *certid* has been revoked. Due to caching, the Dirmngr is able to answer immediately in most cases.

The *certid* is a hex encoded string consisting of two parts, delimited by a single dot. The first part is the SHA-1 hash of the issuer name and the second part the serial number.

Alternatively the certificate's SHA-1 fingerprint *certfpr* may be given in which case an OCSP request is done before consulting the CRL. If the option '`--only-ocsp`' is given, no fallback to a CRL check will be used. If the option '`--force-default-responder`' is given, only the default OCSP responder will be used and any other methods of obtaining an OCSP responder URL won't be used.

Common return values are:

`GPG_ERR_NO_ERROR (0)`

> This is the positive answer: The certificate is not revoked and we have an up-to-date revocation list for that certificate. If OCSP was used the responder confirmed that the certificate has not been revoked.

`GPG_ERR_CERT_REVOKED`
: This is the negative answer: The certificate has been revoked. Either it is in a CRL and that list is up to date or an OCSP responder informed us that it has been revoked.

`GPG_ERR_NO_CRL_KNOWN`
: No CRL is known for this certificate or the CRL is not valid or out of date.

`GPG_ERR_NO_DATA`
: The OCSP responder returned an "unknown" status. This means that it is not aware of the certificate's status.

`GPG_ERR_NOT_SUPPORTED`
: This is commonly seen if OCSP support has not been enabled in the configuration.

If DirMngr has not enough information about the given certificate (which is the case for not yet cached certificates), it will will inquire the missing data:

```
S: INQUIRE SENDCERT <CertID>
C: D <DER encoded certificate>
C: END
```

A client should be aware that DirMngr may ask for more than one certificate.

If Dirmngr has a certificate but the signature of the certificate could not been validated because the root certificate is not known to dirmngr as trusted, it may ask back to see whether the client trusts this the root certificate:

```
S: INQUIRE ISTRUSTED <CertHexfpr>
C: D 1
C: END
```

Only this answer will let Dirmngr consider the CRL as valid.

7.3 Validate a certificate using a CRL

Check whether the certificate with FINGERPRINT (SHA-1 hash of the entire X.509 certificate blob) is valid or not by consulting the CRL responsible for this certificate. If the fingerprint has not been given or the certificate is not know, the function inquires the certificate using:

```
S: INQUIRE TARGETCERT
C: D <DER encoded certificate>
C: END
```

Thus the caller is expected to return the certificate for the request (which should match FINGERPRINT) as a binary blob. Processing then takes place without further interaction; in particular dirmngr tries to locate other required certificate by its own mechanism which includes a local certificate store as well as a list of trusted root certificates.

The return code is 0 for success; i.e. the certificate has not been revoked or one of the usual error codes from libgpg-error.

7.4 Validate a certificate using OCSP

```
CHECKOCSP [--force-default-responder] [fingerprint]
```

Check whether the certificate with *fingerprint* (the SHA-1 hash of the entire X.509 certificate blob) is valid by consulting the appropiate OCSP responder. If the fingerprint has not been given or the certificate is not known by Dirmngr, the function inquires the certificate using:

```
S: INQUIRE TARGETCERT
C: D <DER encoded certificate>
C: END
```

Thus the caller is expected to return the certificate for the request (which should match *fingerprint*) as a binary blob. Processing then takes place without further interaction; in particular dirmngr tries to locate other required certificates by its own mechanism which includes a local certificate store as well as a list of trusted root certificates.

If the option '`--force-default-responder`' is given, only the default OCSP responder is used. This option is the per-command variant of the global option '`--ignore-ocsp-service-url`'.

The return code is 0 for success; i.e. the certificate has not been revoked or one of the usual error codes from libgpg-error.

7.5 Put a certificate into the internal cache

Put a certificate into the internal cache. This command might be useful if a client knows in advance certificates required for a test and wnats to make sure they get added to the internal cache. It is also helpful for debugging. To get the actual certificate, this command immediately inquires it using

```
S: INQUIRE TARGETCERT
C: D <DER encoded certificate>
C: END
```

Thus the caller is expected to return the certificate for the request as a binary blob.

The return code is 0 for success; i.e. the certificate has not been succesfully cached or one of the usual error codes from libgpg-error.

7.6 Validate a certificate for debugging

Validate a certificate using the certificate validation function used internally by dirmngr. This command is only useful for debugging. To get the actual certificate, this command immediately inquires it using

```
S: INQUIRE TARGETCERT
C: D <DER encoded certificate>
C: END
```

Thus the caller is expected to return the certificate for the request as a binary blob.

8 The Client Tool

The `dirmngr-client` is a simple tool to contact a running dirmngr and test whether a certificate has been revoked — either by being listed in the corresponding CRL or by running the OCSP protocol. If no dirmngr is running, a new instances will be started but this is in general not a good idea due to the huge performance overhead.

The usual way to run this tool is either:

```
dirmngr-client acert
```

or

```
dirmngr-client <acert
```

Where *acert* is one DER encoded (binary) X.509 certificates to be tested. The return value of this command is

`0` The certificate under question is valid; i.e. there is a valid CRL available and it is not listed tehre or teh OCSP request returned that that certificate is valid.

`1` The certificate has been revoked

`2 (and other values)`
There was a problem checking the revocation state of the certificate. A message to stderr has given more detailed information. Most likely this is due to a missing or expired CRL or due to a network problem.

`dirmngr-client` may be called with the following options:

`--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.

`--quiet, -q`
Make the output extra brief by suppressing any informational messages.

`-v`

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

`--pem` Assume that the given certificate is in PEM (armored) format.

`--ocsp` Do the check using the OCSP protocol and ignore any CRLs.

`--force-default-responder`
When checking using the OCSP protocl, force the use of the default OCSP responder. That is not to use the Reponder as given by the certificate.

`--ping` Check whether the dirmngr daemon is up and running.

`--cache-cert`
: Put the given certificate into the cache of a running dirmngr. This is mainly useful for debugging.

`--validate`
: Validate the given certificate using dirmngr's internal validation code. This is mainly useful for debugging.

`--load-crl`
: This command expects a list of filenames with DER encoded CRL files. With the option '`--url`' URLs are expected in place of filenames and they are loaded directly from the given location. All CRLs will be validated and then loaded into dirmngr's cache.

`--lookup`
: Take the remaining arguments and run a lookup command on each of them. The results are Base-64 encoded outputs (without header lines). This may be used to retrieve certificates from a server. However the output format is not very well suited if more than one certificate is returned.

`--url`
`-u`
: Modify the `lookup` and `load-crl` commands to take an URL.

`--local`
`-l`
: Let the `lookup` command only search the local cache.

`--squid-mode`
: Run DIRMNGR-CLIENT in a mode suitable as a helper program for Squid's '`external_acl_type`' option.

`dirmngr`(1), `gpgsm`(1)

Appendix A GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

A.0.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

A.0.2 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General

Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.
5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose

any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two

goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

12. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) 19yy  name of author

This program 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 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.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type ‘show w’.  This is free software, and you are welcome
to redistribute it under certain conditions; type ‘show c’
for details.
```

The hypothetical commands ‘`show w`’ and ‘`show c`’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘`show w`’ and ‘`show c`’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program ‘Gnomovision’
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Option Index

Index

G

S

History

- Using DirMngr, 2002, Steffen Hansen, Klarlvdalens Datakonsult AB.
- Using DirMngr, 2004, 2005, 2006, 2008 Werner Koch, g10 Code GmbH.