GSC

version 1.0, 3 June 2007

Sergey Poznyakoff.

Published by the Free Software Foundation, 51 Franklin Street, Fifth Floor
Boston, MA 02110-1301, USA

Short Contents

1 Introduction to GSC 1
2 CVS Tools 3
3 Source Tree Utilities 9
4 Root Utilities 13
5 Sendmail ‘mc’ Files 23
6 Startup Scripts 25
7 User Tools 31
8 How to Report a Bug 35
A GNU Free Documentation License 37
Concept Index 45

Table of Contents

1 Introduction to GSC 1

2 CVS Tools 3
- 2.1 mksnapshot 3
- 2.2 sv_logger 4
- 2.3 Sync WWW 4
  - 2.3.1 sv_sync_www_schedule 4
  - 2.3.2 sv_sync_www 5
    - 2.3.2.1 Single job 5
    - 2.3.2.2 Cron job 6
    - 2.3.2.3 Invocation 6
  - 2.3.3 Seting Up CVS synchronization framework 7
  - 2.3.4 sv_www_loginfo 7

3 Source Tree Utilities 9
- 3.1 fixnamespace 10
- 3.2 fsf-move 10

4 Root Utilities 13
- 4.1 ckaliases 13
- 4.2 bind replication 14
- 4.3 firewall 14
  - 4.3.1 Firewall Primitives 15
- 4.4 Manage PHP Sessions 15
- 4.5 Jabberd 16
  - 4.5.1 Jabberd Operation Overview 16
  - 4.5.2 Jabberd Invocation 17
  - 4.5.3 Jabberd Configuration File 17
    - 4.5.3.1 Configuration File Statements 19
    - 4.5.3.2 An Example of '`jabberd.cfg`' file 20

5 Sendmail '`mc`' Files 23

6 Startup Scripts 25
- 6.1 rc.inet1 25
- 6.2 rc.autofs 26
- 6.3 rc.ntpd 27
- 6.4 rc.local 28
- 6.5 rc.firewall 28

6.6 rc.ipacct 29
6.7 rc.ppp 29
6.8 rc.tagr 30

7 User Tools 31
7.1 dict-setup 31
7.2 consoleconf 32
7.3 ppp 33

8 How to Report a Bug 35

Appendix A GNU Free Documentation License 37
A.1 ADDENDUM: How to use this License for your documents . . . 44

Concept Index 45

1 Introduction to GSC

GSC stands for *Gray's Script Collection*. It is a collection of various mostly unrelated tools and configuration files I use in system administration and software development. The tools are mainly for my own use, however I'd be glad if you find them useful too. The package is distributed under GPL, for exact license terms and conditions see file '`COPYING`' in the main distribution directory.

This manual documents version 1.0 of the package. It is distributed under GFDL, See Appendix A [Copying This Manual], page 37.

2 CVS Tools

This set of tools is designed to facilitate some tasks related to CVS and Savane maintenance.

2.1 mksnapshot

This `mksnapshot` utility creates source tarballs from modules in CVS repository. This is useful, for example, to create daily snapshots from the repository.

The program looks for modules in the `CVS` repository. For each module found, it checks out first line of '`ChangeLog`' file, and if it presents a date later than the one when the last snapshot was made, it checks out the module into a temporary directory, tars its contents and places the resulting archive into the distribution directory.

The resulting archive is named '*project*`-`*DATE*`.tar.gz`', where *project* is the name of the module and *date* is latest '`ChangeLog`' date in '`YYYY-MM-DD`' format.

The program takes a single argument, a name of the configuration file. This file should declare several environment variables that control `mksnapshot` behavior.

`CVSROOT` [Variable]
This variable specifies the CVS root directory name.

`WD` [Variable]
A working directory for extracting CVS data.

`FTP` [Variable]
Target directory where tarballs should be placed.

`MAXSNAPSHOTS` [Variable]
Maximum number of snapshots to keep for each project. This variable is optional. It defaults to 3.

`REPOSITORY` [Variable]
Type of CVS repository. This variable is optional.

Set `REPOSITORY=PLAIN` if you use classic "plain" CVS repository, and `REPOSITORY=SAVANE`, if you use Savane-based repository. Lower-case values are accepted as well. Default is '`SAVANE`'.

The behaviour of `mksnapshot` differs depending on the type of repository used. For plain repositories, `mksnapshot` processes all the modules in the repository, except `CVSROOT`. For Savane repositories, the program first gets the list of user's groups, then for each group it looks up a module of this name. When found, such module is processed as described above. Thus, only those modules are selected, where current user participates.

2.2 sv_logger

This program reads its standard input line by line and sends each line to the given `syslog` priority/facility. It could be used to implement logging in shell scripts, especially those run with non-root privileges.

The program accepts two arguments:

'`-t` *tag*' Specify the syslog *tag* to use. The tag will appear before each logged line. The default tag is '`sv_logger`'.

'`-p` *facility.priority*'
Use specified `syslog` facility and priority. See `syslog.conf(5)` for the list of valid falicity and priority values. The default value is `user.notice`.

2.3 Sync WWW

The programs in this section form a framework for automatic update of a directory upon CVS commit. This is useful, for example, to keep a mirror of a module. Puszcza (http://puszcza.gnu.org.ua) uses this for automatic update of projects' home pages.

This works as follows: suppose that each project keeps publicly available data (html pages, etc) in directory '`/software/`*project*'. These data are under CVS control in directory '`/webcvs/`*project*'. Upon commit, CVS server executes '`CVSROOT/loginfo`' which calls `sv_sync_www_schedule`. This utility schedules an update job. Another program, `sv_sync_www` is called as a `cron` job. This program flushes all jobs scheduled so far. For each project it changes to the project public directory and performs `cvs update` (well, actually it does much more than that... see Section 2.3.2 [sv_sync_www], page 5).

The following subsections describe each utility in detail.

2.3.1 sv_sync_www_schedule

The program `sv_sync_www_schedule` schedules a `CVS` synchronization job for further execution. The jobs are stored in directory '`/var/spool/savane/`'. This directory must have sufficient permissions for `sv_sync_www_schedule` to access it. Normally, this means adding each CVS user to a common group, e.g. '`svusers`' and setting directory access mode to 0775:

```
$ ls -ld /var/spool/savane
drwxrwxr-x    2 root     svusers        25 Aug 18 18:00 savane/
```

The program takes three arguments: user name[1], file name of the CVS repository, and the list of files changed by the commit. The program is supposed to be invoked by each commit. The proper way to do so is to add the following line to the project's '`CVSROOT/loginfo`' file:

[1] not used currently

```
ALL      /bin/sv_sync_www_schedule ${USER} ${CVSROOT} %s
```

`sv_sync_www_schedule` logs errors via `syslog` channel `local1.error`.

2.3.2 sv_sync_www

(This message will disappear, once this node revised.)

The program `sv_sync_www` is the main engine for CVS synchronization. It operates in two modes: in *single job* mode it synchronizes a single CVS commit (called *job*), in *cron job* mode it processes a set of accumulated jobs at once.

Whatever the mode is, the processing of a single job looks as follows: the program determines CVS root directory, subdirectory and files affected by the commit, changes to the directory where the synchronization must be performed (a *destination directory*, and runs `cvs update` with appropriate arguments. After update, `sv_sync_www` recursively searches for files named '`.symlinks`' under the destination directory. Each such file specifies the symbolic links that should be created *within a directory, where it resides*.

In a '`.symlinks`' file any empty lines and lines starting with '`#`' or '`;`' are ignored. Rest of lines must contain two words: the name of the file to create the link from, and the name of the link to be created. Thus, the following `.symlinks` file:

```
# comment
gsc.html index.html
```

will create a link named '`index.html`', pointing to the file '`gsc.html`'.

Notice that the linking is allowed only within or under the directory containing '`.symlinks`' file. In particular, `sv_sync_www` will refuse to create links whose names begin with the directory separator or contain '`..`'.

2.3.2.1 Single job

Single job mode is the default mode for `sv_sync_www`. It operates in this mode, unless '`-s`' command line option is specified.

It takes one or two command line arguments. A single command line argument is parsed as a `%s` parameter in '`loginfo`' invocation, i.e. it must consist of the project directory name and the whitespace separated list of files, affected by the commit. The project directory name can be absent, which is indicated by the leading whitespace. The list of files can have two special forms: '`- New directory`' and '`- Imported sources`'.

In order to invoke `sv_sync_www_flush` on each commit, the '`loginfo`' file must contain a line similar to the following:

```
ALL  (date; cat; (sleep 2; /bin/sv_sync_www %s ) \
        &\ ) >> /var/log/sync_www.log 2>&1
```

There are two notes on this example. First, programs like `sv_sync_www` must be run in the background, otherwise they will cause dead lock conditions due to the use of lock files. Secondly, this example shows only the minimal invocation of `sv_sync_www`. You will always need to pass it the commited

file name or names (see '`%s`' above). However, depending on the actual configuration you may need to pass it some configuration options as well.

2.3.2.2 Cron job

(This message will disappear, once this node revised.)

`Sv_sync_www` starts in cron job mode when it is given '`-s`' command line argument. It takes a single optional argument, specifying the *spool directory* to use (defaults to '`/var/spool/savane`'.

2.3.2.3 Invocation

(This message will disappear, once this node revised.)

First of all, `sv_sync_www` assumes some default settings that you should know about. Namely, it supposes that both CVS repository and update directory (hereinafter called *WWW directory*) are located on the same server. Further, it supposes that the former is called '`/webcvs`' and the latter is called '`/home/puszcza/software`'. And, finally, it invokes `cvs` binary using the following command line:

```
CVS_RSH=ssh cvs -q -z3
```

Of course, any of these defaults can be changed from the command line.

To specify another WWW directory, use '`-w`' option.

To specify a repository directory, use '`-R`'. The argument to this option should be a directory name itself, without any access method prefix. See below for a way to specify this.

In particular, if you use `Savane`-like CVS layout and wish to use immediate synchronization, your '`loginfo`' would contain:

```
ALL  (date; cat; (sleep 2; /bin/sv_sync_www -R ${CVSROOT} %s ) \
        &\ ) >> /var/log/sync_www.log 2>&1
```

The `sv_sync_www_flush` script passes this option by default.

If the repository and WWW directory are located on different servers, the IP address of host name of the server containing WWW directory must be specified via '`-H`'.

In this case `sv_sync_www` will invoke `cvs` binary on the WWW server using `ssh` (more precisely: `/usr/bin/ssh`). The name of the `ssh` binary can be set using '`-r`' option.

Further, invoked `cvs` will need to access *remote* CVS repository. The directory name of this repository can be specified using '`-R`' option, described above. To specify access method, use '`-m`' option. The argument to this option will be prepended to the repository directory (either default one or the one specified using '`-R`' option), so make sure you end it with a colon.

By default, `sv_sync_www` will assume that `cvs` should use `ssh` for accessing the remote server (see the default `cvs` invocation above). If it is not the case, you can use '`-c`' command line option. The `cvs` commands will be appended to the supplied command verbatim.

Finally, please notice that using remote WWW server you will have to set up an access from the CVS to WWW and from the WWW to CVS without passwords (usually this means setting up `ssh` for public keys authentication).

To illustrate these concepts, suppose that the name of the remote WWW server is `www.foo.org`, it should be accessed using `rsh` (Note: *do not use it* in real life. It is unsecure), the repository itself should be accessed using '`gserver`' method, and all CVS interactions should use maximal compression. Then, the command line for `sv_sync_www` will be:

```
sv_sync_www -H www.foo.org -m :gserver:cvs.foo.org: \
            -r rsh -c 'cvs -q -z9' -R ${CVSROOT} %s
```

When setting up unusual configurations, it is often useful to be able to run `sv_sync_www` without actually changing anything. This mode is called *dry run*, and it is turned on by '`-n`' option. In dry run mode, `sv_sync_www` does not change anything, instead it just prints what it would have done. Additional debugging diagnostics can be enabled using '`-x`' option.

Finally, running `sv_sync_www -h` displays a short usage summary.

2.3.3 Seting Up CVS synchronization framework.

Loginfo file

```
ALL  /usr/local/bin/sv_sync_www_schedule ${USER} ${CVSROOT} %s
```

Crontab

```
# Flush WWW synchronisation jobs, scheduled by sv_sync_www_schedule
# once an hour
0 */1 * * *     /usr/local/bin/sv_sync_www -s
```

2.3.4 sv_www_loginfo

The program `sv_www_loginfo` scans '`loginfo`' files of all the projects in the repository and replace the default ones (i.e. the ones whose contents is entirely commented out) with the contents described in the previous subsection. This is to avoid directly modfifying `Savane` sources and to achieve, at the same time, the desired WWW synchronization functionality. Once `Savane` is flexible enough to allow for user-configurable '`CVSROOT`' contents, the need for this program will go away.

At the time of this writing, `sv_www_loginfo` is called from '`crontab`' as follows:

```
30 */2 * * * sv_groups --cron && \
             sv_users --cron && \
             sv_www_loginfo --cron
```

The program assumes the CVS repository under '`/webcvs`'. It takes the following command line options:

'`--cron`' Run as a cron job. All diagnostics is in this case reported via `syslog` channel `local1.info`. Without this option it goes directly to `stderr`.

‘`--help`’ Display short usage summary.

3 Source Tree Utilities

Programs described in this chapter perform a set of global replacements on all the files within a project. The operation is being performed in a separate directory, which is populated by `make distdir` command. After the replacement, `make distcheck` is run to ensure that the operation did not break integrity of the project. If `distcheck` passes successfully, the resulting update can be left in the temporary directory for further inspection, archived in a `tar` archive, or propagated back to the original project.

Prerequisites for running these programs are:

1. Project must conform to GNU Coding Standards. In particular, it must support `make distdir` and `make distcheck` and these two goals must be built successfully on the unmodified project.
2. The program must be run in the project top source directory

The usage synopsis is similar for both programs. The following command line options are supported:

'`-k`' Leave modified project in the temporary working directory. The name of the working directory is formed as *prefix-program*, where *prefix* is the base name of the projects top source directory (unless overridden by '`-p`', see below) and *program* is the name of the invoked program.

'`-r`' Propagate the changed files to the original progect. This option requires GNU `tar` version 1.15.1 or newer.

'`-t`' Store modified project tree in tar archive named '*prefix-program*`.tar.gz`'.

'`-p` *prefix*'
Specify name prefix for temporary working directory and tar archive (see options '`-k`' and '`-t`').

'`-h`' Display short usage summary.

The command line options to `configure` can be given using '`--configure`'. This option can be specified several times, arguments of multiple '`--configure`' options are concatenated in a whitespace separated list. The very first '`--configure`' option should be preceeded by a double-dash:

```
-- --configure --with-prog
```

Any non-option arguments containing an equals sign, and any non-option arguments beginning with double-dash ('`--`') are considered to be additional `make` arguments and are passed to `make` verbatim.

3.1 fixnamespace

`Fixnamespace` utility replaces all global identifiers in a project by prefixing them with a given string. This is useful for creating a common namespace if a project contains a loadable library, hence the name of the utility. Notice that only those symbols are modified, that do not already begin with the prefix.

Global identifiers are any functions, variables and typedefs declared in the header files of the project. To locate these, `fixnamespace` uses *tag files*, created by `etags` utility (see section "Tags Tables" in *GNU Emacs Manual*). Thus, to use `fixnamespace` you must have Emacs version 21.3 or newer installed on your system.

The utility takes two or more arguments. First argument is the prefix to be appended to all symbols. Rest of arguments specify the header files where to look for global symbols. Only file names should be given, without any directory prefixes.

Apart from the command line options common for all source-tree utilities, `fixnamespace` understands following option:

'`-x` *`symbol`*'
: Exclude *symbol* from replacement. This option can be given multiple times.

Example of using this program:

```
fixnamespace -x scm_long2num mu_ message.h mailbox.h body.h list.h
```

Final notice: when too many header files are specified in the command line, the number of created Lisp variable bindings can exceed Emacs capacity; in this case `emacs` will signal an error. To overcome this, increase value of `max-specpdl-size` variable in '`fixnamespace.el`' near line 215.

3.2 fsf-move

During my activity as a free software programmer, the Free Software Foundation has twice changed its locations. Each such move implied a change of postal mail address, and, consequently, a need for updating each file in any free software project, since the standard GPL or LGPL copyright header refers to it.

Such a work can be very tedious for large projects, so when the FSF has recently changed its location again, I decided to automate the process once and for all. This is what `fsf-move` is for.

The program does not take any special command line options or arguments beside those described above (see Chapter 3 [Source Tree Utilities], page 9). Currently it is tuned to change postal mail address from

59 Temple Place, Suite 330 Boston, MA 02110-1301 USA.

to

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Should FSF move again, you will have to change these defaults by modifying '`fsf-move`':177 and '`fsf-move.awk`':21-22.

The program implements a sophisticated search-and-replace algorithm, if it knows a syntax of the language a file is written in. The purpose of the algorithm is to allow for arbitrary splitting of lines in the address and to preserve alignment and formatting of comments. This algorithm is applied to files whose names match following regexps:

*.[chyl] C sources;

.m4|.ac|*.at
M4 sources;

*.sh Shell scripts;

*.awk AWK sources;

.exp|.tcl
TCL sources;

*.am|Makefile.tmpl
Automake sources;

.el|.scm*|*.lisp
Various dialects of Lisp;

PO files (*.po), and their derivatives are ignored.

The rest of files is processed using *fuzzy search* algorithm, which is able to find the postal address in the majority of cases, though not always. Such files should probably be inspected after `fsf-move` finishes its work.

4 Root Utilities

This chapter describes a set of utilities useful in system administration.

4.1 ckaliases

`Ckaliases` checks one or several `sendmail`-style alias files for consistency. Following checks are performed:

1. Transitivity check This check discovers eventual circular dependencies.
2. Use of prohibited aliases

If several alias files are supplied, `ckaliases` treats them as parts of a single alias file.

The program returns 0 if all checks pass successfully. Otherwise, it diagnoses encountered problems and exits with error code 1.

Before processing its input files, `ckaliases` reads definitions of Sendmail `w` class from file '`/etc/mail/sendmail.cw`', or from a file specified using '`-w`' command line option. For example, running

```
ckaliases -w /etc/mail/local-domains aliases
```

instructs `ckaliases` to use '`/etc/mail/local-domains`'.

By default, the program allows any valid Sendmail constructions in its input files. To restrict the input syntax to plain aliases only, i.e. to prohibit use of pipes and file redirections, use '`-r`' command line option. This option affects any file names following it. To cances the restriction, use '`-u`' option. For example:

```
ckaliases aliases -r local -u global
```

This command restricts '`local`' to plain aliases only, while allowing any aliases in files '`aliases`' and '`global`'.

Two options are provided to help trace and debug program's actions. First, '`-v`' increases verbosity level. Secondly, '`-d`' enables debugging mode. This option takes a mandatory argument, a string specifying part or parts of the program to debug. Characters valid in this string are:

l or L Enable lexical analyzer debugging information.

y or Y Enable debugging input grammar analyzer (parser).

Any of these characters can be prefixed with '`-`' (a dash), to disable corresponding debugging feature, e.g. `-d y-l`.

Finally, running `ckaliases -h` displays a short usage summary.

Example of `ckaliases` usage

The following '`/etc/mail/Makefile`' rule constructs '`aliases`' from several input files, provided that these are valid alias files:

```
aliases: .depend Aliases mailman/LISTS /com/mailer/aliases
         ckaliases Aliases -f mailman/LISTS -r /com/mailer/aliases
         @(echo "# WARNING: DO NOT EDIT THIS FILE!!!"; \
         cat Aliases;\
         cat mailman/LISTS | \
         (cd mailman; \
          while read name; \
          do \
             case $$name in \
             \#*) ;; \
             *)   cat $$name;; \
             esac;\
          done);\
         cat /com/mailer/aliases) > aliases
```

4.2 bind replication

(This message will disappear, once this node revised.)

4.3 firewall

The '`firewall`' subdirectory of the source tree contains an `m4` wrapper over GNU/Linux `iptables` utility. The basic idea behind it is to create a framework that could be used without changes on any platform, no matter what the underlying firewall implementation is. Currently it is implemented only for `iptables`. The support for BSD-style `ipfw` and, eventually, more will be added soon.

The framework is based on a set of `m4` wrappers. From the point of view of a final user, its usage is as follows. First, the administrator creates a set of firewall rules using the primitives supplied by '`firewall.m4`'. Then he processes '`firewall.m4`' using `m4` and obtains a set of implementation-dependent firewall rules that can be fed to the shell.

You will probably not need to process '`firewall.m4`' manually, we recommend to use `rc.firewall` instead (see Section 6.5 [rc.firewall], page 28). However, for the sake of completeness, here is how it should be invoked:

```
m4 [-DSYSCONFDIR=dir] -DACTION=action firewall.m4
```

This command generates a set of implementation-dependent firewall rules that can be fed to the shell.

Its arguments are:

`ACTION` [Variable]

Specifies which set of commands `firewall.m4` is to create. Possible values are:

status	List actual firewall rules.
flush	Flush the firewall rules.
filename	Read in and process file '*`filename`*'.

Optional variable `SYSCONFDIR` specifies a directory where '`rule.d`' subdirectory is located (see [rule.d], page 15). It defaults to '`/etc/firewall`'.

4.3.1 Firewall Primitives

(This message will disappear, once this node revised.)

`ACCEPT` *chain, srcip, srcport, dstip, dstport, proto, log* [Firewall Primitive]

`REJECT` *chain, srcip, srcport, dstip, dstport, proto, log* [Firewall Primitive]

`DENY` *chain, srcip, srcport, dstip, dstport, proto, log* [Firewall Primitive]

`LIST` [Firewall Primitive]

`port_setup` *rule ip* [Firewall Primitive]

4.4 Manage PHP Sessions

The script `session-cleanup` provides flexible garbage collection for PHP session files. It is designed first of all for managing session files stored in a deep directory structrure (i.e. when `session.save_path = n;path` in '`php.ini`'), for which the built-in PHP garbage collection is turned off. Beside this, `session-cleanup` allows to set up different *TTL* (*time to live*) intervals for different session files, based on their contents.

The script should be started at regular intervals as a cron job. Upon startup it obtains the value of `session.save_path` variable from '`/etc/apache/php.ini`'. If your '`php.ini`' is located elsewhere, specify its exact location using '`-c`' command line option. Then `session-cleanup` scans the storage directory and removes session files that were created more than TTL days ago. The exact value of TTL is determined using following rules:

1. If the command line contains a non-option argument, it is taken as a name of *configuration file*. This file is scanned for a line that matches the contents of the session file. If such a line is found, it determines the TTL for this session.
2. If '`-t`' is given in the command line, its argument specifies TTL.
3. Otherwise, default value of 3 days is used.

Configuration file has a simple line-oriented syntax. Blank lines and lines that begin with a hash character ('`#`') are ignored. The rest of lines is split into two fields. First field specifies a *regular expression* to search for in the session file. Second field gives the TTL value in days for files that match this regular expression. The first matching line is used.

A sample configuration file follows:

```
# Regex                 TTL (days)
```

```
^gallery_session_.*     1
mprogh                  62
```

If a regular expression contains whitespace, use `shell` escapes or quotes to protect it.

Syntax

```
session-cleanup options [config-file]
```

options are:

'`-c file`' Use *file* instead of '`/etc/apache/php.ini`'.

'`-h`' Display a short help summary.

'`-n`' Dry run. Only display which files will be removed, do not actually remove them.

'`-t ttl`' Set default session time to live. *ttl* is measured in days.

'`-v`' Verbose mode. Useful for debugging in conjunction with '`-n`'.

4.5 Jabberd

The `jabberd` utility is a dispatcher daemon for '`Jabberd 2.x`' (http://www.jabber.org/software/jabberd2x.shtml). It is intended as a replacement for the similar utility shipped with the '`jabberd 2.x`' package. There were two reasons that urged for the replacement: first, the original `jabberd` is written in Perl and consumes way too many resources because of that. Secondly, it is not flexible enough. In particular, it is only able to control jabber daemons, but cannot control external transports (such as GG or GIT.

4.5.1 Jabberd Operation Overview

The GSC `jabberd` daemon reads the list of processes it is supposed to control from its configuration file upon startup. By default, the configuration file is named '`jabberd.cfg`' and located in `$sysconfdir` directory, but its exact location can be overridden at startup (see '`-c`' option, below). If run with the root privileges, `jabberd` switches to the privileges of a selected user (by default '`jabber`') after startup. Then, the program changes file creattion mask to a safe value (default – '`037`'). Unless explicitly requested to remain in the foreground, the utility detaches itself from the controlling terminal and switches to the background. Then, `jabberd` starts the required processes in the order of their appearance in the configuration file. The exact command line options and arguments for each subprocess are specified in the configuration file. If a particular subprocess prints its diagnostics on stderr or stdout, you may instruct `jabberd` to capture it and to divert it to a particular `syslogd` priority (see [stdout], page 19). After launching the subprocesses `jabberd` sleeps until any of them exits. If that happens, the exited subprocess is restarted immediately. If a process is restarted more

than 10 times within a two minutes interval, it is disabled for the next five minutes (the same way the standard UNIX `init` utility operates).

The `jabberd` utility exits if it recieves any of the following signals: SIGTERM, SIGQUIT, SIGINT. It attempts to restart itself if delivered the SIGHUP signal. This is possible only if the utility is started using its absolute file name. In any case, before exiting, the utility shuts down all subordinate processes *in the reverse order* of their appearance in the configuration file. The processes are shut down by sending them SIGTERM signals. If any process does not exit within a 1 second interval, it is re-sent the same signal. This process continues until either all of the processes terminate or the *shutdown timeout* interval expires, whichever happens first. If the latter happens, any processes still left are slayed using SIGKILL signal. The default shutdown timeout is 5 seconds and it may be changed using `shutdown-timeout` configuration file statement (see Section 4.5.3.1 [cfgstat], page 19).

4.5.2 Jabberd Invocation

If started without options, `jabberd` will use the precompiled defaults. Otherwise, the following command line options are understood:

'`-c file`' Use *file* as the main configuration file.

'`-D`' Increase debugging level.

'`-f`' Do not disconnect from the controlling terminal, but run in foreground mode instead. This option is mainly useful for debugging. It implies '`-e`' (see below).

'`-e`' Print all diagnostics on the standard output.

'`-h`' Display a terse usage summary.

'`-p file`' Write master process ID to *file*. This option overrides '`pdifile`' configuration file statement (see the next section).

'`-v`' Print program version and licensing information and exit.

4.5.3 Jabberd Configuration File

The configuration file has a line-oriented syntax. Empty lines are ignored. Comments are introduced by a pound sign ('`#`'), everything starting from the first occurrence of '`#`' up to the end of line is ignored.

Configuration statements consist of *command word* and one or several *arguments*, separated by any amount of whitespace. There are '`simple`' and '`compound`' configuration statements. Simple statements occupy a single line. Compound statements begin with a simple statement, occupy several lines, and end with `end` statement, appearing on a line by itself. Compound statements in turn contain another simple statements.

The simplest working '`jabberd.cfg`' file is:

```
prog router      /usr/local/etc/jabberd/router.xml
prog resolver    /usr/local/etc/jabberd/resolver.xml
prog sm          /usr/local/etc/jabberd/sm.xml
prog s2s         /usr/local/etc/jabberd/s2s.xml
prog c2s         /usr/local/etc/jabberd/c2s.xml
```

This file instructs `jabberd` to launch five basic jabber components and supply the given configuration files to them. The `prog` statement is a simple statement taking one to two arguments. The first one names the program to lauch, and an optional second one specifies its configuration file. All the programs will be launched in the order of their appearance in the '`jabberd.cfg`' file, and will be shut down in the reverse order. The `prog` statement is designed expressly to start jabber core programs, it should not be used to start third-party programs.

To start third-party programs, e.g. transports, use `exec` statement. It is a compound statement that has the following structure:

```
exec tag
 command command-line
 stdout prio
 stderr prio
end
```

The sub-statement `command` specifies the full command line of the program. Notice that most transports behave as daemons. If it is so, you will have to use a special command line option requiring the transport to remain in the foreground (see the transport documentation to find this option). If the program prints its diagnostics on the standard error, the `stderr` statement can be used to capture and redirect it to the syslog. For example, `stderr debug`, instructs `jabberd` to divert the program's standard error to the syslog, using priority '`debug`'. In this case the log entries will be prefixed with *tag*, or, if it is absent, with the first word of *command-line*.

For example, the GG transport can be started using the following statement:

```
exec ggtrans
  command /usr/local/sbin/jggtrans -f
  stdout notice
  stderr notice
end
```

Several configuration statemenst control various aspects of the behavior of the `jabberd`. For example, `user` statement instructs it to switch to privileges of the named user after startup. By default the utility will switch to the privileges of the user '`jabberd`', this statement can be used to change that, for example:

```
user nobody
```

When switching to user privileges, `jabberd` retains only the main user group, as specified in '`/etc/passwd`' file and drops all supplementary groups the user might be a member of. To retain the privileges of a supplementary group, name it with `group` statement. This statement can be used sev-

eral times, to retain several groups. For example, the following statement switches to the privileges of user 'nobody' and retains two supplementary groups: 'staff' and 'ftp':

```
user nobody
group staff
group ftp
```

The following subsubsection describes all configuration file statements in detail.

4.5.3.1 Configuration File Statements

`exec` [*tag*] [Jabber Statement]
Schedule a third-party program (e.g. a transport) for startup. *tag* specifies the prefix to identify this program in the diagnostic poutput. The `exec` statement is a block statement that have the following structure:

```
exec tag
 command command-line
 stdout prio
 stderr prio
 facility fac
end
```

The subordinate statements are:

`command` *command-line* [exec statement]
Set the command line of the transport. *command-line* is parsed much the same way as in shell, except that no variable substitution takes place.

`stdout` *prio* [exec statement]
`stderr` *prio* [exec statement]
Redirect program's standard output (or error, in case of `stderr`) to the given syslog priority. Allowed values for *prio* are: 'EMERG', 'ALERT', 'CRIT', 'ERR', 'WARNING', 'NOTICE', 'INFO', and 'DEBUG', optionally prefixed with 'LOG_'. The string matching is case-insensitive.

`facility` *facility* [exec statement]
This statement does nothing. It is reserved for future use.

`group` *name* [Jabber Statement]
Retain supplementary group *name* after switching to the user's privileges.

`pidfile` *file* [Jabber Statement]
Write master process ID to *file*

`prog` *command* [*config-file*] [Jabber Statement]
Schedule a jabber core program for startup. The program name is given by *command* argument. Optional *config-file* gives the location of its configuration file. The program command line will be (parts enclosed by square brackets being optional):

command [-c *config-file*] [-D]

The '-D' is given only if the jabberd debugging level is greater than 2 (e.g. when running it as jabberd -DDD).

umask *n* [Jabber Statement]

Set file creation mask to *n*. The default umask is '037'.

user *name* [Jabber Statement]

Run with this user privileges.

shutdown-timeout *n* [Jabber Statement]

Wait *n* seconds for all children to shut down.

syslog-facility *facility* [Jabber Statement]

Output diagnostics to the given syslog facility. *facility* is may be one of the following: 'USER', 'DAEMON', 'AUTH', 'AUTHPRIV', 'LOCAL0' through 'LOCAL7', and 'MAIL'. The string matching is case insensitive. Optionally, 'LOG_' prefix may be prepended to *facility*.

syslog-tag *tag* [Jabber Statement]

Mark jabberd diagnostics with the given syslog tag. By default 'jabberd' is used.

4.5.3.2 An Example of 'jabberd.cfg' file

```
# Run as user 'jabber'
user    jabber
# Retain two supplementary groups:
group   staff
group   nobody

# Store PID to the given file
pidfile /usr/local/var/jabberd/pid/jabberd.pid
# Wait 10 seconds for the shutdown of the children.
shutdown-timeout 10

# Start basic jabberd framework:
prog router      /usr/local/etc/jabberd/router.xml
prog resolver    /usr/local/etc/jabberd/resolver.xml
prog sm          /usr/local/etc/jabberd/sm.xml
prog s2s         /usr/local/etc/jabberd/s2s.xml
prog c2s         /usr/local/etc/jabberd/c2s.xml

# Start GG transport and capture its output:
exec ggtrans
  command /usr/local/sbin/jggtrans -f
  stdout notice
  stderr notice
end

# Start ICQ transport and capture its output:
exec jit
```

```
  command /usr/local/bin/jabberd-jit -c /usr/local/etc/jit.xml
  stdout notice
  stderr notice
end
```

5 Sendmail ‘mc’ Files

These are `sendmail` configurations for various machines. To compile them you must have `sendmail` source tree installed.

By default, `configure` will look for `sendmail` source directory in ‘`/usr/src`’ and ‘`/usr/local/src`’. If it finds several `sendmail` versions, it will use the one with the greatest version number.

If the `sendmail` source directory is located elsewhere, specify its exact location with ‘`--with-sendmail-cfdir=`*`dir`*’, for example:

```
./configure --with-sendmail-cfdir=$HOME/sendmail-8.13.1
```

Otherwise, to force `configure` to pick up a specified version of `sendmail`, use ‘`--with-sendmail-version=`*`version`*’ option.

Once the package is configured, you can create all ‘`.cf`’ files using following command:

```
cd mc
make cf
```

To create only ‘*`file`*`.cf`’, run `make` ‘*`file`*`.cf`’.

6 Startup Scripts

This chapter describes several startup files, designed mainly for GNU/Linux. To use any of them, first read its description, then copy it to the location where your startup scripts reside ('`/etc/rc.d`' on Slackware) and make sure it is executed at startup. Then follow the script-specific recommendations set forth in the corresponding section.

6.1 rc.inet1

The startup script `rc.inet1` sets up network interfaces. It was thought as a replacement for Slackware startup script with the same name, which I find extremely inconvenient.

To use the script, copy it to the location where your startup scripts reside ('`/etc/rc.d`' on Slackware) and make sure it is executed at startup. The script itself does not use any configuration files. Instead, it consists of configurable and immutable parts. Both parts are clearly separated with appropriate comments.

Configurable part contains a set of variable definitions. The variables are:

`if_list` [Variable]

This variable contains a whitespace separated list of interface names to be configured. For example:

```
if_list="lo eth0"
```

Each interface name must itself be a valid `shell` variable name. Names of almost all existing interfaces meet this condition. The common exception are ethernet aliases, which on GNU/Linux contain colon. To list such an interface name, replace the colon with underscore. For example to configure interfaces `lo`, `eth0` and `eth0:0`, one would write:

```
if_list="lo eth0 eth0_0"
```

One more note: it is advisable to always keep `lo` in this list.

`if_`*`iface`* [Variable]

For each interface name *iface* in `if_list`, there must be defined a variable named `if_`*`iface`*. The value of this variable must be an `ifconfig` command line initializing given interface. The name of the interface itself should be omitted from the command line. Given the previous example, one might write:

```
# Configure three interfaces
if_list="lo eth0 eth0_0"

# Set up loopback interface
if_lo="127.0.0.1"

# Set up eth0
if_eth0="inet 192.168.10.1 netmask 255.255.255.224\
         broadcast 192.168.10.31"

# Set up eth0:0
if_eth0_0="inet 192.168.10.2 netmask 255.255.255.224\
           broadcast 192.168.10.31"
```

`route_iface` [Variable]

For each interface name *iface* in `if_list`, there can be defined a variable `route_iface`. If it is defined, it contains a `route` command line for setting up routing for this interface. Completing our previous example, for loopback interface one might write:

```
if_lo"127.0.0.1"
route_lo="add -net 127.0.0.0 netmask 255.0.0.0 lo"
```

`default_gw` [Variable]

This variable defines IP address of the default gateway.

`static_routes` [Variable]

The variable `static_routes` declares *route identifiers* for each static route to be set up. A route identifier is an arbitrary word consisting of alphanumeric characters and underscores. For each identifier *id* in `static_routes` there must be declared variable `route_id`, whose value is `route` command line for setting up this route. For example:

```
static_routes="1"
route_1="add -net 192.168.10.0 netmask 255.255.255.224 \
         gw 192.168.10.10 metric 216 reject"
```

6.2 rc.autofs

The script `rc.autofs` controls GNU/Linux automounter facility. It accepts a single mandatory command line argument:

`start` Start automounter.

`stop` Stop automounter.

`status` List configured and active automounter instances and mount points. Here is a sample output (long lines being split for readability):

```
Configured Mount Points:
------------------------
/usr/sbin/automount --timeout=1 \
        /auto file /etc/automount/auto.misc

Active Mount Points:
--------------------
1040 /usr/sbin/automount --timeout=1 \
       /auto file /etc/automount/auto.misc
```

`reload` Reload current configuration, restarting `automount` daemon if necessary.

`restart` Unconditionally restart `automount` daemon.

The script assumes following full file names:

‘`/usr/sbin/automount`’
The name of the `automount` binary. Adjust `DAEMON` variable if it is located elsewhere.

‘`/etc/automount`’
Configuration directory. Adjust `CFGDIR` if you need another name.

‘`/etc/automount/master`’
The name of the master configuration file. Adjust `CFGFILE` variable if you need another name.

‘`/var/lock/subsys/autofs`’
The name of the lock directory. Adjust `LOCKFILE` variable if necessary.

6.3 rc.ntpd

`Rc.ntpd` controls the *network time protocol* daemon. It takes a single mandatory command line argument:

`start` Start `ntpd`. Before starting it the script attempts to synchronize current date with the master servers by running `ntpdate`. If `ntpdate` fails, the script sleeps for a specified number of seconds and attempts to synchronize again. Such attempts are tried several times while increasing proportionally the initial delay between the attempts, so that the delay after *n*th attempt is *n* times the initial one. Two configuration variables are provided to control this behavior: `MAXRETRY` sets the maximum number of attempts, `TIMEOUT` sets the initial delay in seconds. If all attempts fail, `rc.ntpd` sends a mail to root and exits without trying to launch `ntpd`.

`stop` Stop `ntpd`.

`status` List running `ntpd` instances and display current date/time difference between the local machine and its servers. points. Here is a sample output (long lines being split for readability):

```
Running processes:
------------------
1124 /usr/sbin/ntpd

Current date diff:
------------------
server 10.10.10.1, stratum 2, offset -0.008749, delay 0.13557
```

`reload`
`restart` Restart `ntpd` daemon.

The script assumes following full file names:

'`/usr/sbin/ntpd`'
: Name of the `ntpd` daemon. Kept in `DAEMON` variable.

'`/etc/ntp.conf`'
: Main `ntpd` configuration file. Kept in `CONFIG` variable.

The script searches `ntpdate`, `mail` and `ps` binaries using usual path mechanism. You may need to update `PATH` settings at the beginning of the script.

6.4 rc.local

The script `rc.local` starts or stops *local services*. List of the services is kept in the variable `MODLIST`. For each service *s* from that list, `rc.local` attempts to execute script named '`/etc/rc.d/rc.s`'. The argument to `rc.local`, if available, is passed to that string verbatim. If `rc.local` is started without argument, the word start is used instead.

If a service should be started from a non-root user account, prepend its name with *user*`@`. For example, to start `/etc/rc.d/rc.mailman` from user postmaster, set:

```
MODLIST="postmaster@mailman"
```

Finally, if the service name begins with a slash, it is taken as absolute file name, without adding '`/etc/rc.d/rc.`' to it. Although it might be necessary in some cases, such usage is not recommended.

6.5 rc.firewall

This startup script sets firewall using `m4` firewall wrappers (see Section 4.3 [firewall], page 14). It tajes a single mandatory command line argument:

start
restart Set up firewall.

stop Flush all firewall rules.

status List all installed firewall rules.

check
test
: Produce a list of shell commands that will be used to set up firewall rules. Equivalent to `rc.firewall -d start`.

Option '`-d`' can be given before the argument to make `rc.firewall` produce a list of shell commands it would have used to perform a given task. For example

```
rc.firewall -d start
```

prints commands that would be executed by `rc.firewall start`.

The script searches for `m4` wrappers in '`/etc/firewall`'. This location can be changed by setting `FWDIR` environment variable.

The list of firewall rules is expected to be located in '`$FWDIR/rules.m4`'. This location can be changed by setting `START` environment variable.

6.6 rc.ipacct

This script controls the `ipacct` daemon (). It takes a single mandatory command line argument indicating what action to take:

start
: Start the daemon.

stop
: Stop the daemon.

status
: Display PID of the current instance of `ipacct`.

restart
: Unconditionally restart the daemon.

reload
reconfigure
: Tell `ipacct` to re-read its configuration file.

The script searches for `ipacct` binary in the current `PATH`. Two variables defined at the beginning of the script control its behavior:

`IFACE` [Variable]
: Interface where to install `ipacct`. Default is eth0.

`PIDFILE` [Variable]
: Location of `ipacct` pid file. Default is '`/var/run/ipacct-IFACE.pid`'.

6.7 rc.ppp

This script sets up a set of PPP connections at start up. It reads configuration from the file '`/etc/ppp.conf`'. The configurations is expressed as a set of shell variables.

`ppp_list` [Variable]
: This variable contains a whitespace separated list of *identifiers* for PPP connections. Each identifier is an arbitrary word, such that it is a valid shell variable. For each identifier *id*, the configuration file must declare a shell variable `ppp`*`id`*`_options`, specifying command line options for `pppd`.

`ppp`*id*`_options` [Variable]
This variable specifies command line options for connection *id*.

A sample '`ppp.conf`' file follows:

```
ppp_list="0 1"
ppp0_options="modem passive nodetach defaultroute crtscts \
 lock 192.168.0.1:192.168.0.2 /dev/ttyS0 9600"
ppp1_options="modem lock 192.168.0.1:192.168.0.3 /dev/ttyS1 57600"
```

6.8 rc.tagr

`Rc.tagr` controls `tagr` daemon (). It takes a single mandatory command line argument indicating what action to take:

start — Start the daemon.

stop — Stop the daemon.

status — Display PID of the current instance of `tagr`.

restart — Unconditionally restart the daemon.

reload
reconfigure
Tell `tagr` to re-read its configuration file.

The script searches for `tagr` binary in '`/usr/local/sbin/tagr`' and for its PID file in '`/var/run/tagr.pid`'. This can be changed by editing variables `PROGRAM`, `PIDFILE` and `CMD` at the beginning of the script.

7 User Tools

7.1 dict-setup

`Dict-setup` automates setting up a set of `aspell` dictonaries (see section "Overview" in *GNU Aspell Manual*). It takes a list of file names as its arguments. Each file must contain a list of languages for which spell checking dictionaries must be installed. The languages should be specified as ISO 639 language abbreviations, one per line. Empty lines and shell-style comments are ignored.

For each language, `dict-setup` downloads its latest spell checking dictionary from `ftp://ftp.gnu.org.ua/gnu/aspell/dict`, builds and installs it.

If you wish to use another URL, specify it via '`-u`' command line option, for example

```
dict-setup -u ftp://ftp.gnu.org FILE
```

To save bandwidth, `dict-setup` looks for dictionaries compressed via `bzip2` command. This can be altered by setting `SUF` variable to the desired file suffix, e.g. `SUF=.tar.gz`.

By default, `dict-setup` installs the latest dictionary version. If this behavior is not desired, you can select which dictionary to install by invoking the program with '`-c`' option.

When run with this option, `dict-setup` prints a list of available dictionaries for each language and allows you to select one for download (it uses `lynx` to display the list, so make sure it is in your `PATH`). Use following commands to navigate through the list:

⟨v⟩ View index file.

⟨x⟩ Do not install dictionaries for this language. Proceed to the next language.

⟨q⟩ Abort installation immediately.

⟨*number*⟩ Install *number*th dictionary from the list.

Following is the complete list of `dict-setup` options:

'`-c`' Run in confirm mode: for each language, present the user with a list of available dictionaries and allow him to select which one to install.

'`-h`' Print a short usage summary and exit.

'`-v`' Verbosely list each action executed.

'`-u` *url*' Use *url* instead of the default `ftp://ftp.gnu.org.ua/gnu/aspell/dict`.

7.2 consoleconf

`Consoleconf` sets console encoding and input method, for both text and X consoles. The program uses `kbd` package (ftp://ftp.win.tue.nl/pub/linux-local/utils/kbd/).

`Consoleconf` consists of the following parts:

1. `consoleconf` binary. It is normally installed as `$prefix/consoleconf`.
2. Data directory with *language definition* files. It is normally installed as '`$datadir/consoleconf`'.

Usage of `consoleconf` is quite simple. Running:

```
consoleconf lang
```

sets up the console for language *lang*, i.e. it sets keyboard map, optional screen map and screen font. Command line option '`-c`' can be used to specify encoding. For example:

```
consoleconf -c latin1 no
```

The package is shipped with support for the following languages:

da	Danish
es	Spanish
el	Greek
no	Norwegian
pl	Polish
ru	Russian
uk	Ukrainian

To add a new language, create its *language description file* describing how to set up the console for this language. I suggest to use two-letter language codes as per ISO 639. The file has a regular `sh`(1) syntax. It should define following variables:

`Consoleconf variable` *KEYMAP* [Variable]
Specifies the name of the keymap to use. Keymaps are usually stored in '`/usr/share/kbd/keymaps/`', although the exact location may vary.

`Consoleconf variable` *FONT* [Variable]
Specifies the name of console font file to use when in text console. Console font files are usually located in '`/usr/share/kbd/consolefonts/`', although the exact location may vary.

`Consoleconf variable` *X11* [Variable]
`Setxkbmap` command line to use when under X11.

`Consoleconf variable` *MOTD* [Variable]
Message of the day. It is displayed after the console is set up. If it starts with commercial at sign, it is taken as a name of file

whose contents is the message of the day. The file is looked up in '`$datadir/consoleconf/motd`'.

This variable is optional.

For example, the following is the definition of Polish language:

```
KEYMAP=pl.map
FONT="-m 8859-2 iso02.16.gz"
X11="-rules xfree86 -model pc102 -layout pl"
```

7.3 ppp

This sub-package provides a framework for setting up a dial-up connection. At the core of it is `start-ppp`, a program which actually installs a connection. Various configuration data are supplied to it by several configuration files, normally located in '`/etc/ppp`'. When run, `start-ppp` interprets the file '`/etc/ppp/pppscript.m4`' to determine various configuration settings for the modem. There is normally no need to edit this file, as the actual configuration data are stored in two files it includes: '`/etc/ppp/modem`' and '`/etc/ppp/login`'.

Modem configuration is kept in file '`/etc/ppp/modem`'. This file should define the following variables.

`MODEM_INITSTRING` [Variable]
Initialization string for the modem.

`MODEM_TIMEOUT` [Variable]
Default modem timeout in seconds.

`MODEM_DIALPREFIX` [Variable]
Dial prefix, '`t`' for tone dialing, '`p`' for pulse dialing

`AREA_CODE` [Variable]
Optional area code.

This file defines dial-up user credentials. These are kept in two variables:

`LOGIN` [Variable]
Specifies user login name.

`PASSWORD` [Variable]
Specifies user password.

Finally, the list of numbers to dial is supplied by file '`/etc/ppp/numbers`'. This file must contain one number per line. Empty lines and shell-style comments are allowed. The numbers from this file are tried in turn until dialing one of them succeeds or the end of file is reached. By default, `start-ppp` first looks up the file '`numbers`' in current working directory. If it is found, it is read instead of '`/etc/ppp/numbers`'.

8 How to Report a Bug

As I said, this is not a package on its own. Rather `gsc` is a collection of scripts for my own use. However, I will be pleased if you will find it useful, and even more pleased if you will send me your suggestions or bug reports. If you decide to do so, write to gray@gnu.org.ua.

As the purpose of bug reporting is to improve software, please be sure to include maximum information when reporting a bug. The mimimum information needed is:

- Topmost date from the '`ChangeLog`' file.
- Conditions under which the bug appears.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within

that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque

copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If

there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire

aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) *year* *your name*.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.2
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled ‘‘GNU
> Free Documentation License’’.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

> with the Invariant Sections being *list their titles*, with
> the Front-Cover Texts being *list*, and with the Back-Cover Texts
> being *list*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Concept Index

This is a general index of all issues discussed in this manual

-

'--configure', fixnamespace and fsf-move command line option 9
'--with-sendmail-cfdir' 23
'--with-sendmail-version' 23
'-c' command line option, dict-setup. 31
'-c' command line option, session-cleanup 15
'-c', sv_sync_www option 6
'-d', ckaliases option 13
'-h' command line option, dict-setup. 31
'-h' command line option, session-cleanup 16
'-h', ckaliases option 13
'-h', sv_sync_www option 7
'-H', sv_sync_www option 6
'-k', fixnamespace and fsf-move command line option 9
'-m', sv_sync_www option 6
'-n' command line option, session-cleanup 16
'-n', sv_sync_www option 7
'-p', fixnamespace and fsf-move command line option 9
'-r', ckaliases option 13
'-r', fixnamespace and fsf-move command line option 9
'-r', sv_sync_www option 6
'-R', sv_sync_www option 6
'-t' command line option, session-cleanup 15
'-t', fixnamespace and fsf-move command line option 9
'-u' command line option, dict-setup. 31
'-u', ckaliases option 13
'-v' command line option, dict-setup. 31
'-v' command line option, session-cleanup 16
'-v', ckaliases option 13
'-w', ckaliases option 13
'-w', sv_sync_www option 6
'-x', sv_sync_www option 7

.

'./numbers' 33
'.symlinks' 5

/

'/etc/ppp/login' 33
'/etc/ppp/modem' 33
'/etc/ppp/numbers' 33
'/etc/ppp/pppscript.m4' 33

A

ACCEPT 15
ACTION 14
AREA_CODE 33

C

CFGDIR, rc.autofs configuration variable 27
CFGFILE, rc.autofs configuration variable 27
ckaliases 13
ckaliases, '-d' command line option 13
ckaliases, '-h' command line option 13
ckaliases, '-r' command line option 13
ckaliases, '-u' command line option 13
ckaliases, '-v' command line option 13
ckaliases, '-w' command line option 13
ckaliases, example 13
CMD 30
command 19
CONFIG, rc.ntpd configuration variable 28
configuration file for session-cleanup 15
consoleconf 32
Consoleconf variable 32
consoleconf, adding new language 32
consoleconf, available languages 32
consoleconf, invocation 32
consoleconf, usage 32
CVS snapshots, creating 3
CVS synchronization 4
CVS synchronization job, scheduling ... 4

CVSROOT 3

D

DAEMON, rc.autofs configuration variable 27
DAEMON, rc.ntpd configuration variable 28
default_gw 26
DENY 15
dict-setup 31
dict-setup command line options 31

E

exec 19

F

facility 19
FDL, GNU Free Documentation License 37
firewall 14
Fixing a project namespace. 10
fixnamespace 10
fixnamespace, command line options 9
fixnamespace, command line options. 10
fixnamespace, invocation 10
fsf-move 10
fsf-move, command line options 9
FTP 3
FWDIR 29

G

group 19
GSC defined 1

I

if_*iface* 25
if_list 25
IFACE 29

J

jabberd 16

L

LIST 15
LOCKFILE, rc.autofs configuration variable 27
LOGIN 33
'loginfo', invoking sv_sync_www 5
'loginfo', setting up for real time CVS synchronization 5
'loginfo', setting up for scheduled CVS synchronization 4
lynx 31

M

MAXRETRY, rc.ntpd configuration variable 27
MAXSNAPSHOTS 3
mksnapshot 3
MODEM_DIALPREFIX 33
MODEM_INITSTRING 33
MODEM_TIMEOUT 33
MODLIST, rc.local configuration variable 28

P

PASSWORD 33
pidfile 19
PIDFILE 29, 30
port_setup 15
ppp_list 29
ppp*id*_options 30
ppp*id*_options, rc.ppp configuration variable 30
prog 19
PROGRAM 30

R

rc.autofs 26
rc.firewall 28
rc.inet1 25
rc.ipacct 29
rc.local 28
rc.ntpd 27
rc.ppp 29
rc.tagr 30
REJECT 15
REPOSITORY 3
route_*id*, rc.inet1 configuration variable 26
route_*iface* 26

S

Scheduling a CVS synchronization job .. 4
Session TTL 15
session-cleanup 15
`session-cleanup`, configuration file ... 15
`session-cleanup`, setting session TTL 15
`shutdown-timeout` 20
snapshots, creating for CVS 3
`START` 29
`start-ppp` 33
`static_routes` 26
`stderr` 19
`stdout` 19
`SUF` 31
`sv_logger` 4
`sv_sync_www` 5
`sv_sync_www`, accessing remote WWW server 6
`sv_sync_www`, command line options 6
`sv_sync_www`, cron job mode 6
`sv_sync_www`, debugging 7
`sv_sync_www`, default settings 6
`sv_sync_www`, dry run 7
`sv_sync_www`, invoking from '`loginfo`' .. 5
`sv_sync_www`, obtaining help summary .. 7
`sv_sync_www`, single job mode 5
`sv_sync_www`, specifying CVS command. 6
`sv_sync_www`, specifying remote repository access method 6
`sv_sync_www`, specifying remote shell binary 6
`sv_sync_www`, specifying repository directory 6
`sv_sync_www`, specifying update server name 6
`sv_sync_www`, specifying WWW directory 6
`sv_sync_www_schedule` 4
`sv_www_loginfo` 7
`syslog-facility` 20
`syslog-tag` 20

T

`TIMEOUT`, `rc.ntpd` configuration variable 27

U

`umask` 20
`user` 20

W

`WD` 3