1 GNU Rot[t]log

This is the GNU Rot[t]log info manual version 0.72 GNU Rot[t]log is the GNU log management utility. It was originally authored in 2000 by Stefano Falsetto to rotate his logs. It has similar syntax to Red Hat's logrotate(8), and is written in **bash.**

GNU Rot[t]log is designed to simplify administration of systems that generate large numbers of log files. It allows automatic rotation, compression, and archiving of logs. It also mails reports to the system administrator. Each log file may be handled daily, weekly, monthly, in user-defined days, or when it becomes too large.

Normally, rottlog is run as a daily cron job. The logfiles cannot be modified multiple times per period unless the `-f` or `--force` option is used.

Version 0.72

1.1 Synopsis and Options

rottlog [options]

`--checkrc <rcfile>,<rcfile>,...`
: Check syntax of specified configuration file(s). <*rcfile*> must be one, or more, of: *daily, weekly, monthly, all.*

`--help`
: Display help page and exit.

`--version`
: Display version, disclaimer, and exit.

`-d`
: Display default options that are assumed, even if not explicitly used, and exit.

`-l logfile`
: Show contents of archived *logfile* with default pager. *logfile* can be an absolute, or relative, filename. If *logfile* is a relative filename, it will be searched in `$packdir` directory.

`-p program"`
: Use *program* for paging output of showlog. Useful only with `--showlog`. rottlog relies on system *pager* variable by default.

`-s logfile`
: Display a summary of syntax statistics about *logfile*. It is like `--checkrc` but more verbose.

`-f [<daily>] [<weekly>] [<monthly>]`
: Force rotation and archiving of logfiles, even if unnecessary. This might be useful after adding new entries to a configuration file. If daily, weekly, or monthly is not used, the program will force rotation and archiving of all entries in all config files. Using `--force` option will force rotation and archiving of specified config file, and will check for rotation and archiving of remaining config files that follows in time.

`-v`
: rottlog STDERR verbosity. Remember that crontab sends its default output via email.

1.2 Mandatory RC Variables

The main config file is named `rc` and is the first file read by rottlog. Each line can be defined with a variable and default value, `variable=value`. Some variables do not need a value, `variable`. Variables are split in two categories: mandatory and optional. `rc` can be overridden by "period-related" config files.

Comments may appear anywhere in the config file as long as the first non-whitespace character on the line is a `#`. Example:

```
# This is a valid comment
logpart 100b # This is an INVALID comment
```

Following are the mandatory variables needed to make rottlog to work:

compress=<options>
extension=<ext>
fromuser=<sender address>
mail=<mailer prg> (1)
maxdepth=<depth>
nomail (1)
packdir=<archive-dir>
packer=<absolute-filename>
pager=<pager-program>
touser=<receiver address>
uncompress=<options>
unpacker=<absolute-filename>

(1) Can't be used together

Where:

`compress=<options>`
: This variable defines options to be used with the packer program to compress data

  IMPORTANT: The packer program sends output to STDOUT, (ie for gzip/bzip2 use `-c` flag).

`extension=<ext>`
: This variable is used to expand meta-variable `@`*COMP_EXT*, gz for gzip, bz2 for bzip2, etc.

`fromuser=<sender address>`
: Default FROM: address in mail messages sent after each rotation.

`mail=<mailer prg>`
: Mail program to send messages. Example:

  - for sendmail use: `mail="sendmail -t"`
  - for qmail use: `mail="qmail-inject -h"`

`maxdepth=<depth>`
: Default maximum depth to search for files when using * glob in log filename definition block.

`nomail`
: Don't mail rottlog's activity report. This variable is used if the 'mail' option won't be used.

`packdir=<archive-dir>`
: Default basedir to store rotated and archived files. This variable is used to expand `@`*DEF_DIR* meta-variable.

`packer=<absolute-filename>`
: Complete filename to be used for compressed logs.

`pager=<pager-program>`
: Filename of program to use with `--showlog` parameter. It's used in a pipe, so it must be a program able to read from STDIN. The `--showlog` option will not be available if this variable is not defined, and each time rottlog is called, a warning message will be printed.

`touser=<receiver address>`
: Recipient address of mail messages sent by "fromuser".

`uncompress=<options>`
: Options to be used with unpacker program.

  IMPORTANT: Unpacker program must output data to STDOUT. (For gunzip/bunzip2 use -c.)

`unpacker=<complete-filename>`
: Complete filename to be used for uncompressed logs.

1.3 Optional RC Variables

Following are optional variables:

createdir
default_storefile=<storefile>
dir_grp=<group>
dir_own=<owner>
dir_perm=<permissions>
follow_symlinks
ifempty
maxage=<days>
missingok
nocompress
nocreate
nomissingok
notifempty
remove_missing
SunMon=<sun|mon>
tabooext=<list>

Where:

`createdir`
: Create directories specified in storedir if needed. It's useful to use with meta-vars. See Section 1.6 [Use of meta-variables], page 15.

`default_storefile=<storefile>`
: Default filename to use with archived androtated logfiles. It's useful to use with meta-vars. See Section 1.6 [Use of meta-variables], page 15.

  WARNING: Time-related meta-vars are NOT compatible with rotate parameter.

  If `default_storefile` is defined using `@`*WEEK*, `@`*DAY*, `@`*MONTH*, or `@`*YEAR* meta-variables, each config file containing a logfile to be rotated will not be valid.

`dir_grp=<group>`
: Default group of newly created directory. Must exist in `/etc/group` file. Use with createdir option.

`dir_own=<owner>`
: Default owner of newly created directory. Must exist in `/etc/passwd` file. Used with createdir option.

`dir_perm=<permissions>`
: Default permission of newly created directory. Only numeric notation is permitted. Used with createdir option.

`follow_symlinks`
: Follow symlinks to logfiles. Default behaviour is to not follow symlinks.

`ifempty`
: Rotate logfiles even if they are empty.

`maxage`
: Remove archived logs older than <days> days. The age is only checked if the logfile is to be managed.

`missingok`
: If a logfile is missing, go to the next one without issuing an error message. Last action, if defined, will be performed. Post rotate script, if defined, will not be performed.

`nocompress`
: Don't compress logfiles to be archived and rotated.

`nocreate`
: Don't create a new filename. Can't be used with logpart option.

`nomissingok`
: rottlog will return an error message and stop execution if the a file is missing.

`notifempty`
: Do not rotate a logfile if it is empty.

`remove_missing`
: rottlog will stop execution if a delayed file no longer exists or is not readable.

`SunMon=<sun|mon>`
: Use Sunday or Monday as first day of week. This option is useful only in conjunction with @WEEK meta-variable.

`tabooext=<list>`
: The list of files expanded with * glob will not include files with extensions contained in <list>. If a + precedes this list of extensions, the current taboo extension list is augmented, otherwise it is replaced. At startup, the taboo extension list contains: .new .rpmorig, .rpmsave, ,v, .swp, .rpmnew, and ~.

1.4 Global period file structure

Global structure of a period-config file is:

```
out-of-block-definitions

configuration-block-1-start {
  block-contents
}

configuration-block-2-start {
  block-contents
}
```

Comments may appear anywhere in the config file as long as the first non-whitespace character on the line is a #.

At the beginning of all the period related config files (exactly before, and out of, all configuration blocks) all options from the main rc config file can be used. So it can define period-related default values and override its values inside per-logfile block definitions.

Here is another out-of-block definition:

```
include CONFIGFILE[,CONFIGFILE,...]
```

This option includes specified files as if they were written at the end of configuration file. Nested includes are not supported. The *configfile* must be an absolute filename, or a filename with the * wildcard. To include all files in a directory, the search path must be used `path-to-dir/*` (i.e. `/etc/rottlog/monthly.d/*`). If this line is too long, a \ character can be inserted at the end of the line, continuing on the following line.

A `configuration-block-n-start` is a list of one, or more, comma separated filenames to be archived, or rotated. A configuration block for a set of filenames, all stored in a single dir, can be used: `path-to-dir/*` (i.e. `/var/adm/apachelogs/*`). If a filename contains special characters like `]` or `@{` you must protect them from expansion with a double slash (i.e. apache\\[1\\].log). If the same logfile is used more than one time in the same configuration file, it will be used in only the first defined block.

All expanded filenames will be checked to not have an extension contained in the *def_taboo_ext* variable. See [tabooext], page 14.

1.5 Configuration parameters

A configuration block can include one or more of following keywords:

append-only
collate logfile
create mode owner group
createdir [<perms>] [<own> <grp>]
create_logrotate
delaycompress
dateext
dateoffset
day_based
firstaction ... endaction
ifempty
lastaction ... endaction
logpart <qdldl>
log_rotate
mailopt <what>
maxage <days>
maxdepth <depth>
missingok
month_based
nocompress
nocreate
nomail
nosharedscripts
nostoredir
notifempty
period <period-definitions>
postrotate ... endscript
prerotate ... endscript
rotate number
sharedscripts
size num[b,k,M]
start <number>
storedir <dirname>
storefile <filename>
tabooext <list>
tarcollate <logfile>
touser <email address>
week_based
year_based

Where:

`append-only`
: To be used only on ext[23] fs. Set append only attribute to a logfile. (More precisely: remove this attribute before handling log, and reset it after all operations are preformed).

`collate`
: Store more than one logfile in one compressed file. This option makes possible to weekly rotate a logfile (i.e. in a small volume), and monthly store a compressed file containing previously rotated files (i.e. in a bigger backup volume).

If used in monthly configuration file, *logfile* will be searched in weekly configuration file. If used in weekly configuration file, *logfile* will be searched in daily configuration file. This parameter can't be used in daily configuration file.

Logfiles to be referenced with collate option must use rotate option with correct periods:

- collate monthly a weekly rotated file : rotate 4

- collate weekly a daily rotated file : rotate 7

Example: Collect monthly a weekly rotated log.daemon

In weekly configuration file:

```
/var/adm/log.daemon {
  storedir /mnt/tiny/@YEAR/@BASENAME
  createdir 4700 stefano root
  create 600 stefano root
  prerotate
      /sbin/killall -HUP syslogd
  endscript
  # Important!
  rotate 4
  delaycompress
}
```

In monthly configuration file:

```
/var/adm/log.daemon {
  storedir /mnt/BigVolume/Collections/@YEAR/@BASENAME
  createdir
  collate
}
```

In /mnt/BigVolume/Collections/2002/log.daemon will be stored a file called log.daemon.<number> containing log.daemon.1, log.daemon.2, log.daemon.3 and log.daemon.4 weekly created in the /mnt/tiny/2002/log.daemon dir.

This option is compatible only with following:

- [no]storedir
- storefile
- createdir
- [no]missingok (TODO)
- firstaction/lastaction

- prerotate/postrotate
- nocompress
 touser
- nomail
- ifempty/notifempty are silently ignored.

`create [[mode] [owner group]]`

Immediately after rotation, and before the postrotate script is run, the log file is created with the same name as the log file just rotated. *mode* specifies the mode for the log file in octal notation (the same as chmod(2)), *owner* specifies the user name who will own the log file, and group specifies the *group* the log file will belong to.

If none of the three parameters are specified, and `create_logrotate` is set to 0 (or undef), the new file will be created with permissions and owner defined in default variables `fil_perm`, `fil_own`, and `fil_grp`. You can specify *mode* only, or *mode* and *owner*, but you can't specify *mode* and *group* (second parameter is always interpreted as owner). If none of the three parameters are specified, and `create_logrotate` is set to 1, the new file will be created with permissions and owner of old logfile.

`createdir [perms] [own grp]`

Create (if needed) a directory to store archived or rotated logs with specified permissions and owner:group. If none of the three parameters is specified the directory will be created with permissions and owner defined in default variables *dir_perm*, *dir_own*, and *dir_grp*. You can specify *perms* only, or *perms* and *own*, but you can't specify *perms* and *grp* (second parameter is always interpreted as owner). *perms* specifies the mode for the log file in octal notation (the same as chmod(2)).

IMPORTANT: If directory hierarchy to be created is deeper than one directory, only the last directory will have the owner and permissions expected. Example:

```
storedir 2002/04/apache_logs
createdir 4300 rottlog
```

If dir 2002 is not already present on your box you will have (assuming that rottlog is run by user stefano):

```
$ ls -od 2002 2002/04 2002/04/apache_logs
drwxr-xr-x  2 stefano 1024 Apr 11 18:35 2002/
drwxr-xr-x  2 stefano 1024 Apr 11 18:35 2002/04
d-ws------  2 rottlog 1024 Apr 11 18:35 2002/04/apache_logs
```

`create_logrtate`

Create a new 0-size file with permission of just-rotated logfile. This is a parameter used for compatibility with logrotate. **IMPORTANT:** it needs `create`.

`delaycompress`

Postpone compression of the previous log file until the next rotation cycle. It can be used when some program can not be told to close its logfile and thus might continue writing to the previous log file for some time.

`dateext`

Archive old versions of log files, adding a daily extension, like YYYYMMDD, instead of simply adding a number. This is a shortcut for: `storefile @BASENAME.@NEXT_EXT-@YEAR@MONTH@DAY`

`dateoffset` *<+/-day>*

Force time related metavariables to be calculed with <day> days of offset. Example: if today is 5 August, and "dateoffset -1" is used, the @DAY metavariable will be expanded as 04, not 05.

`day_based`

Force `@NEXT_EXT` generation to be on a daily basis. This is useful only if `@DAY` metavariable is used in storefile. See See [month_based], page 10.

`firstaction` *EXIT_CODE...* `endaction`

The unique line between firstaction and endaction (both of which must appear on lines by themselves) are executed before any operation. It will be performed during parameter parsing, before filename and metavars expansion. Optional parameter *EXIT_CODE* is the default exit code expected at the action. If not specified 0 will be expected. If used in a multiple files definition, the action will be run only one time, before processing the first file.

`ifempty`

Rotate the log file even if it is empty.

`lastaction` *EXIT_CODE...* `endaction`

The unique line between lastaction and endaction (both of which must appear on lines by themselves) are executed after all operations. It will be performed just before handling the next logfile. Optional parameter *EXIT_CODE* is the default exit code expected by the action. If not specified 0 will be expected.

If used to define multiple files, the action will be run only one time, after processing all files.

`logpart` *qdldl*

Leave a part of log to archive/rotate in new logfile. *qdldl* can be:

<number>l
<number>b
"<regexp>"
"<#1 day>"

where `l` and `b` stands for lines and bytes respectively. `regexp` is a regular expression used by grep to find the first line to leave in log. `"#1 day"` is a shortcut to a regexp corresponding to the first day of the current month.

`log_rotate`

Allows rottlog to use a different algorithm to rotate logs. In particular it will use the logrotate algorithm. This is a parameter used for compatibility with logrotate.

`mailopt` *what*

Allows rottlog to send detailed mail messages. *what* can be a valid combination of:

all|a	⇒	Mail all sorts of information
maillast|overwrite|over	⇒	Attach to-be-overwrited files to mail in plain text
ziplast|zip	⇒	Attach to-be-overwrited files to mail in compressed format
error|err	⇒	Mail only error messages
none|nomail	⇒	Don't mail anything

`maxage` <*days*>

Remove archived logs older than <days> days. The age is only checked if the logfile is to be managed. If maxage is used without <days>, the default is 365 days.

`maxdepth` <*depth*>

Default maximum depth to search for files when using the * metachar in log filename definition.

`missingok`

If the log file is missing, go on to the next one without issuing an error message. Default behaviour is to stop executing and return an error exit code.

`month_based`

Force `@NEXT_EXT` generation to be on a monthly basis. This is useful only if the `@MONTH` metavariable is used in storefile.

Example:

```
$ date
Fri Feb 25 14:34:04 CET 2005
$ ls
apache.log-2005-01.1
apache.log-2004-02.1
apache.log-2004-08.1
$ rottlog
$ ls
apache.log-2005-01.1
apache.log-2004-02.1
apache.log-2004-08.1
apache.log-2005-02.2
```

In the above example, month_based is used to order apache.log files on a month basis:

```
$ ls *-02*
apache.log-2004-02.1
apache.log-2005-02.2
```

`nocompress`

Old versions of log files are not compressed with packer defined in rc file.

`nocreate`

Don't create a new logfile. Can't be used with logpart.

`nomail`
: Don't send notification email message.

`nosharedscripts`
: Run prerotate and postrotate scripts for every script which is rotated (this is the default, and overrides the sharedscripts option).

`nostoredir`
: Logs are rotated/archived in the same directory the log normally resides in.

`notifempty`
: Do not rotate the log if it is empty.

`period` *`period-definitions`*

WARNING: This option can be used only in custom configuration file.

This is a very powerful option able to force an action on the time interval specified. This is useful when a system administrator must force rotation of some logfiles in times different from the canonical monthly, weekly, and daily.

period-definitions is a comma-separated list of items defining a day, or a set of days, a month, or a set of months, and so on. Usable items can be:

DD	⇒	Each DDth day of the month
DDd	⇒	Each DD days
WWw	⇒	Each WW weeks
mmM	⇒	Each mm months
<weekday>	⇒	Each <weekday> of the week
<monthname>	⇒	Each <monthname> of the year
HH:MM	⇒	At specified hour and minutes (*)
0	⇒	Each time rottlog is called (*)

(*) It depends on rottlog's entries in crontab

Where

```
<weekday> can be short:
    mon, tue, wed, thu, fri, sat, sun
 or long:
    monday, tuesday, wednesday, thursday, friday,
    saturday, sunday

<monthname> can be short:
    jan, feb, mar, apr, may, jun, jul, aug, sep,
    oct, nov, dec
 or long:
    january, february, march, april, may, june,
    july, august, september, october, november,
    december
```

Following characters can add use of logical functions and ranges:

, (comma)	⇒	OR
`" "` (white space)	⇒	AND

! (exclamation mark)	⇒	NOT
- (minus)	⇒	Week or month range
+ (plus)	⇒	Week or month list (Inlined OR)

Minus character can be used to define a range of days or months, can't use long and short mixed range definitions, (i.e. it's valid to use sep-dec but not september-dec).

While the minus character is used to define a range of days or months, plus char can be used to define a list of days or months. In fact it can be used as a shortcut to comma. See below for more information.

The NOT operator must be used before an operator to negate. In a range, or list, a not operator can only be used before its definition. (i.e. it's valid to write `!mon-thu`, but it's not valid to write `mon-!thu`. It's valid to write `!mon+fri+sat`, but it's not valid to write `mon+!fri+sat`)

Comma-separated tokens are analyzed in the given order. When one meets a true condition, the check is stopped without continuing to parse further.

WARNING: Correct behaviour of rottlog with period parameter strictly depends from correct configuration of rottlog's crontab.

Some examples:

```
1) period mon-wed 21:00,sat 10:00
```

Is equivalent to:

```
   period mon 21:00,tue 21:00,wed 21:00,sat 10:00
```

This means that rotation must be performed if rottlog is run and today is Monday and it is 21:00 hours, or today is Tuesday and it is 21:00 hours, and so on with Wednesday, or today is saturday and it is 10:00 hours.

```
2) period !mon-wed 21:00,wed 19:00
```

Is equivalent to:

```
   period !mon !tue !wed 21:00,mon 19:00
```

This means that rotation must be performed if rottlog is run and today is not Monday, Tuesday, nor Wednsday and it is 21:00 hours, or today is Monday and it is 19:00 hours.

```
3) period jan-jun 23:00,jul-sep 10:00,oct-dec 23:00
```

Is equivalent to:

```
   period !jul-sep 23:00,jul-sep 10:00
```

This means that rotation must be performed if rottlog is run and today is not in a month between (or including) July and September and it is 23:00 hours, or today is in a month between (or including) July and September and it is 10:00 hours.

```
4) period mon+fri jul-sep 22:00,
          !jul-sep mon-sat 01:00 (all in one line)
```

Is equivalent to:

```
  period mon jul-sep 22:00, fri jul-sep 22:00,
```

`!jul-sep mon-sat 01:00 (all in one line)`

This means that rotation must be performed if rottlog is run and today is Monday or Friday in a month between (or including) July to September and it's 22:00, or today is not in a month between (or including) July to September and it's between (or including) Monday through Saturday and it's 01:00.

This sample can be made to use the `+` operator. It's handy not to have to write a long line like the latter, but a more compact line like the former is better.

`postrotate ... endscript`

Lines between postrotate and endscript (both of which must appear on lines by themselves) are executed after the log file is archived/rotated. Can be used with metavariables. See Section 1.6 [Use of meta-variables], page 15, for details.

`prerotate ... endscript`

Lines between prerotate and endscript (both of which must appear on lines by themselves) are executed before the log file is archived/rotated. Can be used with metavariables. See Section 1.6 [Use of meta-variables], page 15, for details.

`rotate` *`number`*

Log files are rotated `<number>` times before being overwritten. Rot[t]log uses two rotation methods: a rottlog method and the logrotate method. The native method will order rotated files by extension. The logrotate method can be used adding the log_rotate option to the configuration block. The following example will show the differences between the two methods:

```
# In weekly configuration file
/var/http/logs/access.log {
  rotate 4
  log_rotate
}
/var/http/logs/error.log {
  rotate 4
}
```

The following are saved as specified above:

week 1:	access.log	⇒	access.log.1
	error.log	⇒	error.log.1
week 2:	access.log.1	⇒	access.log.2
	access.log	⇒	access.log.1
	error.log	⇒	error.log.2
week 3:	access.log.2	⇒	access.log.3
	access.log.1	⇒	access.log.2
	access.log	⇒	access.log.1
	error.log	⇒	error.log.3
week 4:	access.log.3	⇒	access.log.4
	access.log.2	⇒	access.log.3

access.log.1	⇒	access.log.2
access.log	⇒	access.log.1
error.log	⇒	error.log.4

After four weeks, the first archived access.log will be access.log.4 and first archived error.log will still be error.log.1

`sharedscripts`

Let the postrotate script be run only once, after handling the last file in a multiple file definition block:

```
/var/adm/log.* {
  sharedscripts
  postrotate
    killall -HUP syslogd
  endscript
}
```

The killall command will be run only after handling all log files beginning with "log" in the /var/adm dir.

`size` *`num`* `[b,k,M]`

Rotate a logfile only if it grows the specified dimension. Parameter must be a number followed by one of following chars:

b - means bytes
k - means kbytes (num * 1024)
M - means megabytes (num * 1000000)

If none of these three characters is used, b is default.

`start` *`number`*

Number to use as the base for rotation. Extensions will start from the specified `number`. The files will be rotated the number of times specified in the rotate option, minus the 'start' number.

`storedir` *`dirname`*

If `dirname` is a relative dirname it will be appended to packdir, defined in the rc file; if `dirname` is an absolute dirname it will be used instead of packdir. Can be used with metavariables. See Section 1.6 [Use of meta-variables], page 15, for details.

`storefile` *`filename`*

Filename of compressed logfile to be used (filename only). Can be used with metavariables. See Section 1.6 [Use of meta-variables], page 15, for details.

`tabooext <list>`

The list of files expanded with the * wildchar will not include files with an extension contained in <list>. If a + precedes this list of extensions, the current taboo extension list is augmented, otherwise it is replaced. At startup, the taboo extension list contains: .new .rpmorig, .rpmsave, ,v, .swp, .rpmnew, and ~.

`tarcollate`
: Same as collate, but store files in a compressed tar file. In addition this option is compatible with ifempty option too. If set, a void file will be archived with tar. If not set, it will be skipped.

  Tar archive will contain relative dirnames if possible (if archived, files are in subdirs of packdir defined in main rc file), or absolute dirname.

`touser` *`e-mail`*
: Override global value of touser variable defined in rc file.

`week_based`
: Force `@NEXT_EXT` generation to be on a week basis. This is useful only if `@WEEK` metavar is used in storefile. See See [month_based], page 10, for more information.

`year_based`
: Force `@NEXT_EXT` generation to be on a yearly basis. This is useful only if `@YEAR` metavar is used in storefile. See See [month_based], page 10, for more information.

1.6 Use of meta-variables

In monthly/weekly/daily logs metavariables can be used. Metavariables are identifiers of a runtime defined value and can be used while using following parameters:

storedir
storefile
prerotate...endscript
postrotate...endscript

List of handled metavariables:

@DIRNAME
: Dir containing logfile to be archived/rotated.

@FILENAME
: Name of the file to be rotated with numerical extension (`@BASENAME.@NEXT_EXT`).

@BASENAME
: Name of the file to be rotated, without extension.

@DEF_DIR
: Default dir of archive dir (corresponding to packdir variable).

@COMP_EXT
: Extension of compressed file (corresponding to extension variable).

@NEXT_EXT
: Next available numbered extension in destination dir.

@TEMPDIR
: Name of a temporary directory.

@1..@n String corresponding to first, second, etc, on directory in dirname of file to archive/rotate

@YEAR Return current 4 digit year (i.e. 1970).

@MONTH Return current month (01..12).

@WEEK Return number of week in year (0..53) to SunMon variable.

@DAY Return current day of month (01..31).

Values of metavariables are defined while the script is running, so a block definition like this

```
/var/adm/log.* {
  storedir /var/adm/archive-log/@BASENAME
  rotate 5
  ...
}
```

means that rottlog will rotate 5 times all files in /var/adm beginning with "log." and store each file in a dir named with logname. (i.e. /var/adm/archive-log/log.kern, /var/adm/archive-log/log.daemon...)

WARNING: Metavariables substitution is merely textual!

So in a block like this

```
/var/http/www.mysite.com/log/access.log {
  storefile @3-@13-@FILENAME
  rotate 4
  ....
}
```

rottlog will create logfiles called something like www.mysite.com-var3-access.log.1

Because `@13` is not defined (complete absolute filename in first line of block definition contains only 5 tokens) and only `@1` will be substituted. Following text will not be affected (remaining 3-) and will **NOT** generate an error code.

1.7 RC File Examples

Example 1. Basic configuration for main rc file. Rottlog will use gzip/gunzip to compress/uncompress data. MTA used is sendmail, with administrator contact email of master@example.net. Rotation algorithm and use of file creation attribute will be the same as logrotate. For security, symbolic links will not be followed.

```
# This file has been created for maximum compatibility with logrotate
#
# Compressor program
packer=gzip

# Packer's options to compress and send to stdout
compress="-9c"
```

```
# DE-Compressor program
unpacker=gunzip

# Decompressor's options to send output do stdout
uncompress="-c"

# Normal extension of compressed files
extension="gz"

# Pager used to view uncompressed logs
pager="less"

# default basedir to store rotated/archived files
packdir="/var/log/rottlog"

# Sender of mail messages
fromuser="rottlog@example.net"

# Receiver of mail messages
touser="master@example.net"

# don't rotate/archive files of 0 size
notifempty=1

# Mail program to send messages.
mail="/usr/sbin/sendmail -t"

# Default Maximum depth to use with * metachar
maxdepth=0

# set to 0 will not follow symlinks
follow_symlinks=0

# Silently remove from status file no longer existant delayed logfiles
remove_missing

# Store archived logfiles in the same directory where "live" logfiles
# resides
nostoredir

# Copy mode, owner, and group from old logfile, if create is called
# without parameters (logrotate behaviour)
create_logrotate

# Rotate files with logrotate behaviour
```

```
log_rotate
```

1.8 Period Related File Examples

Using Meta Variables and a rotation algorithm it is possible to rotate logs so that a file rotated once is never touched again (unless deleted), making log rotation much more compatible with host-based intrusion detection schemes. Words beginning by `@` are special Meta Variables used to create dynamic fields like file and directory names to use.

The following examples will use defaults defined in the previous section. These examples are valid in all monthly, weekly, and daily config files.

Example 1. Use `/var/log/rottlog/log.daemon` and leave in new logfile events from first day of actual month (logpart). After that, a script will be called to restart syslogd (postrotate/endscript).

```
/var/log/rottlog/log.daemon {
   postrotate
      /sbin/killall -HUP syslogd
   endscript
   logpart "#1 day"
}
```

Example 2. Same as Example 1, but archived logfile compression is postponed to the next time rottlog will handle `/var/log/rottlog/log.daemon` (delaycompress).

```
/var/log/rottlog/log.daemon {
   postrotate
      /sbin/killall -HUP syslogd
   endscript
   logpart "#1 day"
   delaycompress
}
```

Example 3. Split log files; postrotate script will be run for each single logfile.

```
/var/log/rottlog/log.daemon,/var/log/rottlog/log.debug {
   postrotate
      /sbin/killall -HUP syslogd
   endscript
   logpart "#1 day"
   delaycompress
}
```

Example 4. Rotate 6 times all files in `/var/log/apache` dir (`rotate`). Archived logfiles will not be compressed (`nocompress`) and they will be stored in an apache-archives subdir of "packdir" defined in main rc configuration file (`storedir`). If a logfile has size 0 it will not be handled (`notifempty`).

```
/var/log/apache/* {
   storedir apache-archives
   rotate 6
   notifempty
   nocompress
}
```

Example 5. Archive `wtmp` and `lastlog` files and touch new 0 byte files with specified permissions, owner and group (`create`). Report message will be sent to a different user from those specified in main rc config file (`touser`). Logfile compression is postponed to the next rotation cycle.

```
/var/adm/wtmp,/var/adm/lastlog {
        create 644 root root
        delaycompress
        touser "admin@example.net"
}
```

Example 6. Rotate twice all logfiles in `/var/log` called `log.a*`, descending two levels of recursion to find files. Store compressed logfiles in a dir called something like:

```
$packdir/<actual year>/<actual month>/<logfile basename>
```

This is done using META-VARIABLES in storedir parameter.

See Section 1.6 [Use of meta-variables], page 15, for details.

New 0 byte files will be created with specified permissions, owner and group (`create`), and will be set with the append-only attribute (`append-only`). Don't rotate logs if they are smaller than 1 Megabyte (`size`).

```
/var/log/log.a* {
     # Descend two levels of depth to find files respecting criteria
     # (beginning by log.a)
     maxdepth 2

     # Use of meta-variables. storedir will be expanded for each processed
     # file (so i.e. will be used 2002/04/log.auth, 2002/04/log.apache, ....)
     # to store compressed archived logs
     storedir @YEAR/@MONTH/@BASENAME

     # Make new dirs if necessary with specified permissions, owner and group
     # ALERT: See README for more details.
     createdir 0640 root loggers

     # Rotate files with a 6 month period.
     rotate 6

     # Flag to use only with an ext2 filesystem. Add append-only attribute
     # to logfile
```

```
    append-only

    # Don't rotate if logfile is smaller than 1 Megabyte
    size 1M
}
```

Example 7. Archive all logfiles stored in local `/usr/local/apache/logs` dir in a remote NFS volume, mounted during rottlog's execution. Will be used `firstaction`...`endaction` to mount nfs remote volume, and `lastaction`...`endaction` to umount it. Archived logfiles are stored in a dir available only after firstaction is performed and is defined during rottlog's execution.

```
/usr/local/apache/logs/* {
  # Action between firstaction and endaction tags will be performed before
  # all logfiles are rotated/archived
  firstaction
    mount fserver.example.net:/LogArchive /mnt/LogVol
  endaction

  # Define a destination directory available only after firstaction
  # is performed
  storedir /mnt/LogVol/@YEAR/@MONTH/@BASENAME

  # Permissions about newly created dir
  createdir 0640 root loggers
  notifempty

  # After each log file is rotated following commands will be executed
  postrotate
    /usr/local/apache/bin/apachectl restart
    /usr/local/bin/my-nice-script.sh
  endscript

  # Action between lastaction and endaction tags will be performed after
  # all logfiles has been rotated/archived.
  lastaction
    umount /mnt/LogVol
  endaction
}
```

Example 8. Archive all logfiles stored in the local `/usr/local/squid/logs/` directory onto tape. Before archive the logfile, it will be saved in a temporary directory that will be removed immediately after handled last logfile in this block.

```
/usr/local/squid/logs/* {
  # Action between firstaction and endaction tags will be performed
  # before all logfiles are rotated/archived
  firstaction
```

```
    mt -f /dev/nst0 eom
  endaction

  # Define a temporary storedir
  storedir @TEMPDIR

  # Define filename of logs to be archived
  storefile @FILENAME.@WEEK@YEAR

  # Don't archive file if it's empty
  notifempty

  # After each log file is rotated following commands will be executed
  postrotate
    star -cv -f /dev/nst0 @TEMPDIR/@FILENAME.@WEEK@YEAR
  endscript

  # Action between lastaction and endaction tags will be performed after
  # all logfiles has been rotated/archived
  lastaction
    mt -f /dev/nst0 offline
  endaction
}
```

Example 9. Archive `log.daemon` only on 15th day of the month, and will compress the archived logfile next month.

```
/var/adm/log.daemon {
        # These actions will be performed after archived each logfile
        postrotate
            /sbin/killall -HUP syslogd
        endscript
        delaycompress
        # rottlog will handle this file only on 15th day of the month
        period 15
}
```

Example 10. Rotate `fetchmail.log` five times before overwriting, each three days, So the file `fetchmail.log.1` will be overwrited each 15 days.

```
/var/log/fetchmail.log {
        # Handle this file every three days
        period 3d
        # Handle logfile on a 5-period basis. So fetchmail.log.1 will be
        # overwritten every 15 days
        rotate 5
}
```

Example 11. Archive `log.auth` using different behaviour in different year periods. If we are in summer (july to september), `log.auth` will be archived on monday or friday at 22:00. If we are not in summer, the file will be archived from monday to saturday at 01:00.

```
/var/adm/log.auth {
        # rottlog will handle this file:
        # monday or friday on 22:00 during summer (july to september)
        # from monday to saturday on 01:00 otherwise
        period mon+fri jul-sep 22:00, !jul-sep mon-sat 01:00
}
```

Example 12. Rotate `log.auth` with logrotate algorithm.

```
/var/adm/log.auth {
  log_rotate
  rotate 4
}
```

Example 13. Rotate `log.auth` using logrotate algorithm and rottlog create parameter. This means that the new logfile will be created with 0600 mode and owner stefano, group root.

```
/var/adm/log.auth {
  log_rotate
  create 600 stefano root
  rotate 4
}
```

Example 14. Rotate `log.auth` using logrotate algorithm and create parameter. This means that the new logfile will be created with same permission and owner.group of just-rotated `log.auth`.

```
/var/adm/log.auth {
  log_rotate
  create_logrotate
  rotate 4
}
```

Example 15. Store apache logs from many virtual hosts each odd days if it is not Summer, otherwise it will store logs only Wednesday and Saturday

```
/hosts/domain1.com/log/access_log,/hosts/domain1.com/log/error_log,\
/hosts/domain2.com/log/access_log,/hosts/domain2.com/log/error_log,\
/hosts/domain3.com/log/access_log,/hosts/domain3.com/log/error_log,\
/hosts/domain4.com/log/access_log,/hosts/domain4.com/log/error_log,\
/hosts/domain5.com/log/access_log,/hosts/domain5.com/log/error_log {
# @1 stays for: first token in path parsing
# @2 stays for: second token in path parsing
storedir @1/@2/log/@MONTH-@YEAR

# Create new directories if not yet existant with specified
```

```
    # permissions, owner and group
    createdir 0644 apache apache

    # Defines archived filenames
    storefile @BASENAME.@DAY.gz

    # Create new 0-size logfiles in place of archived with specified
    # permissions, owner and group
    create 0644 apache apache

    # Force archiving of logfiles:
    # 1 - Monday, wednesday, friday and sunday if actual month is not
    #     between july and august
    # 2 - Wednesday and saturday if actual month is between july and
    #     august
    period !jul-aug mon+wed+fri+sun, jul-aug wed+sat

    # Handle this file even if it is empty
    ifempty

    # Don't mail to administrator report for each log file handled
    nomail
  }
```

Example 16.

```
/var/adm/messages {
   # Store compressed messages in $packdir/messages
   storedir messages
   # These actions will be performed before archiving the logfile
   prerotate
   /sbin/killall -STOP myprogram
   endscript
   # Rotate logs with extensions from .1 to .5
   rotate 5
   # Don't rotate log if it's empty
   notifempty
}
```

1.9 Files

@ETCDIR/rc
: The main configuration file.

@ETCDIR/monthly
: Configuration file to handle monthly log rotation.

@ETCDIR/weekly
: Configuration file to handle weekly log rotation.

@ETCDIR/daily
Configuration file to handle daily log rotation.

@ETCDIR/custom
Configuration file to handle custom log rotation.

@STATDIR
/var/lock/LOCK.rottlog
Lockfile to prevent running multiple instances of rottlog.

1.10 Bugs

If you find a bug in this program, or documentation, please email `bug-rottlog@gnu.org`, or through Bug Tracker at `https://savannah.gnu.orgbugs?group=rottlog`.

1.11 GFDL

Version 1.3, 3 November 2008

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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

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 in 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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

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 1.3
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.

1.12 Mandatory Variables

1.13 Index of meta-variables

1.14 Index of concepts