GNU sharutils, version 4.13.3

A set of shell archiver utilities
Edition 4.13.3, 31 December 2012

Jan Djärv
François Pinard
Bruce Korb

This manual documents version 4.13.3 of the GNU shar utilities.

Table of Contents

GNU `shar` makes so-called shell archives out of many files, preparing them for transmission by electronic mail services, while `unshar` helps unpacking shell archives after reception. Other tools help using `shar` with the electronic mail system, and even allow synchronization of remote directory trees. This is release 4.13.3.

This manual documents version 4.13.3 of the GNU shar utilities.

1 Introduction to this toolset

GNU `uuencode` and `uudecode` have an history which roots are lost in ages, and we will not even try to trace it. The current versions were brought into GNU by Ian Lance Taylor, and later modernized by Ulrich Drepper. GNU `shar` surely has a long history, too. All along this long road, numerous users contributed various improvements. The file `THANKS` in the distribution, as far as we know, contain the names of all contributors we could identify, and for which email addresses are seemingly valid.

Please help us getting the history straight, for the following information is somewhat approximative. James Gosling wrote the public domain `shar 1.x`. William Davidsen rewrote it as `shar 2.x`. Warren Tucker implemented modifications and called it `shar 3.x`. Richard Gumpertz maintained it until 1990. François Pinard, from the public domain `shar 3.49`, made `GNU shar 4.x`, in 1994. Some modules and other code sections were freely borrowed from other GNU distributions, bringing this `shar` under the terms of the GNU General Public License.

Your feedback helps us to make a better and more portable product. Mail suggestions and bug reports (including documentation errors) for these programs to `bug-gnu-utils@prep.ai.mit.edu`.

2 The basic shar utilities

GNU **shar** makes so-called shell archives out of many files, preparing them for transmission by electronic mail services. A *shell archive* is a collection of files that can be unpacked by /bin/sh. A wide range of features provide extensive flexibility in manufacturing shars and in specifying shar *smartness*. For example, shar may compress files, uuencode binary files, split long files and construct multi-part mailings, ensure correct unsharing order, and provide simplistic checksums. See Section 2.1 [shar Invocation], page 3.

GNU unshar scans a set of mail messages looking for the start of shell archives. It will automatically strip off the mail headers and other introductory text. The archive bodies are then unpacked by a copy of the shell. unshar may also process files containing concatenated shell archives. See Section 2.2 [unshar Invocation], page 16.

2.1 Invoking shar

If no files are specified, the list of input files is read from a standard input. Standard input must not be a terminal. shar creates "shell archives" (or shar files) which are in text format and can be emailed. These files may be unpacked later by executing them with /bin/sh. The resulting archive is sent to standard out unless the -o option is given. A wide range of features provide extensive flexibility in manufacturing shars and in specifying shar "smartness". Archives may be fairly simple (--vanilla-operation) or essentially a mailable tar archive.

Options may be specified in any order until a file argument is recognized. If the --intermix-type option has been specified, more compression and encoding options will be recognized between the file arguments.

Though this program supports uuencode-d files, they are deprecated. If you are emailing files, please consider mime-encoded files. If you do uuencode, base 64 is the preferred encoding method.

This section was generated by **AutoGen**, using the agtexi-cmd template and the option descriptions for the shar program. This software is released under the GNU General Public License, version 3 or later.

2.1.1 shar help/usage (--help)

This is the automatically generated usage text for shar.

The text printed is the same whether selected with the help option (--help) or the more-help option (--more-help). more-help will print the usage text by passing it through a pager program. more-help is disabled on platforms without a working fork(2) function. The PAGER environment variable is used to select the program, defaulting to more. Both will exit with a status code of 0.

```
shar (GNU sharutils) - create a shell archive - Ver. 4.13.3
USAGE:  shar [ -<flag> [<val>] | --<name>[{=| }<val>] ]... [file ...]


specifying compression:

   -p, --intermix-type        specify compression for input files
```

```
                                - prohibits these options:
                                vanilla-operation options:
   -C, --compactor=str        specify compaction (compression) program
                                - prohibits these options:
                                vanilla-operation options:
                                - may appear multiple times
   -g, --level-of-compression=num pass LEVEL for compression
                                - It must be in the range:s
                                  1 to 9t be in the range:s
       --level-for-gzip=num   This is an alias for 'level-of-compression'

specifying file encoding methodology:

   -M, --mixed-uuencode       decide uuencoding for each file
   -B, --uuencode             treat all files as binary
                                - an alternate for mixed-uuencode
   -T, --text-files           treat all files as text
                                - an alternate for mixed-uuencode

specifying file selection and output modes:

   -o, --output-prefix=str    print output to file PREFIX.nn
   -l, --whole-size-limit=num split archive, not files, to size
                                - requires these options:uuencode
                                output-prefixese options:uuencode
                                - is scalable with a suffix: k/K/m/M/g/G/t/T
                                - It must lie in one of the ranges:M/g/G/t/T
                                  8 to 1023, orn one of the ranges:M/g/G/t/T
                                  8192 to 4194304one of the ranges:M/g/G/t/T
   -L, --split-size-limit=num split archive or files to size
                                - requires these options:he ranges:M/g/G/t/T
                                output-prefixese options:he ranges:M/g/G/t/T
                                - is scalable with a suffix: k/K/m/M/g/G/t/T
                                - It must lie in one of the ranges:M/g/G/t/T
                                  8 to 1023, orn one of the ranges:M/g/G/t/T
                                  8192 to 4194304one of the ranges:M/g/G/t/T
                                - an alternate for whole-size-limitM/g/G/t/T
   -I, --input-file-list=str  read file list from a file

Controlling the shar headers:

   -n, --archive-name=str     use name to document the archive
   -s, --submitter=str        override the submitter name
   -a, --net-headers          output Submitted-by: & Archive-name: headers
                                - requires these options:size-limitM/g/G/t/T
                                archive-namehese options:size-limitM/g/G/t/T
   -c, --cut-mark             start the shar with a cut line
   -t, --translate           translate messages in the script

Protecting against transmission issues::

       --no-character-count  do not use ‘wc -c’ to check size
   -D, --no-md5-digest       do not use md5sum digest to verify
   -F, --force-prefix        apply the prefix character on every line
   -d, --here-delimiter=str  use delim to delimit the files

Producing different kinds of shars::

   -V, --vanilla-operation   produce very simple shars
   -P, --no-piping           use temporary files between programs
   -x, --no-check-existing   blindly overwrite existing files
   -X, --query-user          ask user before overwriting files
                               - prohibits these options:ize-limitM/g/G/t/T
                               vanilla-operation options:ize-limitM/g/G/t/T
   -m, --no-timestamp        do not restore modification times
   -Q, --quiet-unshar        avoid verbose messages at unshar time
   -f, --basename            restore in one directory, despite hierarchy

Internationalization options::

       --no-i18n             do not internationalize
       --print-text-domain-dir  print directory with shar messages

user feedback/entertainment:

   -q, --quiet               do not output verbose messages
       --silent              This is an alias for ’quiet’

version, usage and configuration options:

   -v, --version[=arg]       Output version information and exit
   -h, --help                Display extended usage information and exit
   -!, --more-help           Extended usage information passed thru pager
   -R, --save-opts[=arg]     Save the option state to a config file
   -r, --load-opts=str       Load options from a config file
                               - disabled as --no-load-optse-limitM/g/G/t/T
                               - may appear multiple timesse-limitM/g/G/t/T

Options are specified by doubled hyphens and their name or by a single
hyphen and the flag character.

If no ‘‘file’’s are specified, the list of input files is read from a
standard input.  Standard input must not be a terminal.
```

```
The following option preset mechanisms are supported:
 - reading file $HOME/.sharrc

``shar'' creates "shell archives" (or shar files) which are in text format
and can be emailed.  These files may be unpacked later by executing them
with ``/bin/sh''.  The resulting archive is sent to standard out unless the
``-o'' option is given.  A wide range of features provide extensive
flexibility in manufacturing shars and in specifying ``shar'' "smartness".
Archives may be fairly simple (``--vanilla-operation'') or essentially a
mailable ``tar'' archive.

Options may be specified in any order until a ``file'' argument is
recognized.  If the ``--intermix-type'' option has been specified, more
compression and encoding options will be recognized between the ``file''
arguments.

Though this program supports ``uuencode''-d files, they are deprecated.  If
you are emailing files, please consider mime-encoded files.  If you do
``uuencode'', base 64 is the preferred encoding method.

please send bug reports to:  bug-gnu-utils@gnu.org
```

2.1.2 compression options

specifying compression.

intermix-type option (-p).

This is the “specify compression for input files” option.

This option has some usage constraints. It:

- must not appear in combination with any of the following options: vanilla-operation.

Allow positional parameter options. The compression method and encoding method options may be intermixed with file names. Files named after these options will be processed in the specified way.

compactor option (-C).

This is the “specify compaction (compression) program” option. This option takes an argument string `PROGRAM`.

This option has some usage constraints. It:

- may appear an unlimited number of times.
- must not appear in combination with any of the following options: vanilla-operation.

The `gzip`, `bzip2` and `compress` compactor commands may be specified by the program name as the option name, e.g. `--gzip`. Those options, however, are being deprecated. There is also the `xz` compactor now. Specify `xz` with `-C xz` or `--compactor=xz`.

Specifying the compactor "'none'" will then disable file compression, even for non-text files. Any other compactor will force uuencoding of files, and the recipient must have uudecode to unpack the archive. (Compressed files are never processed as plain text.)

Specifying the compactor compress is deprecated.

level-of-compression option (-g).

This is the "pass level for compression" option. This option takes an argument number LEVEL. Some compression programs allow for a level of compression. The default is 9, but this option allows you to specify something else. This value is used by gzip, bzip2 and xz, but not compress.

bzip2 option (-j).

This is the "bzip2 and uuencode files" option.

This option has some usage constraints. It:

- may appear an unlimited number of times.

bzip2 compress and uuencode all files prior to packing. The recipient must have uudecode bzip2 in order to unpack.

NOTE: THIS OPTION IS DEPRECATED

gzip option (-z).

This is the "gzip and uuencode files" option.

This option has some usage constraints. It:

- may appear an unlimited number of times.

gzip compress and uuencode all files prior to packing. The recipient must have uudecode and gzip in order to unpack.

NOTE: THIS OPTION IS DEPRECATED

compress option (-Z).

This is the "compress and uuencode files" option.

This option has some usage constraints. It:

- may appear an unlimited number of times.
- must be compiled in by defining HAVE_COMPRESS during the compilation.

compress and uuencode all files prior to packing. The recipient must have uudecode and compress in order to unpack.

NOTE: THIS OPTION IS DEPRECATED

level-for-gzip option.

This is an alias for the level-of-compression option, see [shar level-of-compression], page 7.

bits-per-code option (-b).

This is the “pass bits (default 12) to compress” option. This option takes an argument string BITS.

This option has some usage constraints. It:

- must be compiled in by defining HAVE_COMPRESS during the compilation.

This is the compression factor used by the compress program.

NOTE: THIS OPTION IS DEPRECATED

2.1.3 encoding options

specifying file encoding methodology. Files may be stored in the shar either as plain text or uuencoded. By default, the program selects which by examining the file. You may force the selection for all files. In intermixed option/file mode, this setting may be changed during processing.

mixed-uuencode option (-M).

This is the “decide uuencoding for each file” option.

This option has some usage constraints. It:

- is a member of the mixed-uuencode class of options.

Automatically determine if the files are text or binary and archive correctly. Files found to be binary are uuencoded prior to packing. This is the default behavior for shar.

For a file to be considered a text file instead of a binary file, all the following should be true:

1. The file does not contain any ASCII control character besides BS (backspace), HT (horizontal tab), LF (new line) or FF (form feed).
2. The file contains no character with its eighth-bit set.
3. The file contains no line beginning with the five letters "‘from ’", capitalized or not. (Mail handling programs will often gratuitously insert a > character before it.)
4. The file is either empty or ends with a LF (newline) byte.
5. No line in the file contains more than 200 characters. For counting purpose, lines are separated by a LF (newline).

uuencode option (-B).

This is the “treat all files as binary” option.

This option has some usage constraints. It:

- is a member of the mixed-uuencode class of options.

Use uuencode prior to packing all files. This increases the size of the archive. The recipient must have uudecode in order to unpack. Compressed files are always encoded.

text-files option (-T).

This is the "treat all files as text" option.

This option has some usage constraints. It:

- is a member of the mixed-uuencode class of options.

If you have files with non-ascii bytes or text that some mail handling programs do not like, you may find difficulties. However, if you are using FTP or SSH/SCP, the non-conforming text files should be okay.

2.1.4 in-out options

specifying file selection and output modes.

output-prefix option (-o).

This is the "print output to file prefix.nn" option. This option takes an argument string `PREFIX`. Save the archive to files `prefix.01` thru `prefix.nn` instead of sending all output to standard out. Must be specified when the `--whole-size-limit` or `--split-size-limit` options are specified.

When *prefix* contains a '`%`' character, *prefix* is then interpreted as a `sprintf` format, which should be able to display a single decimal number. When *prefix* does not contain such a '`%`' character, the string '`.%02d`' is internally appended.

whole-size-limit option (-l).

This is the "split archive, not files, to *size*" option. This option takes an argument number `SIZE`.

This option has some usage constraints. It:

- is a member of the whole-size-limit class of options.
- must appear in combination with the following options: output-prefix.

Limit the output file size to `size` bytes, but don't split input files. If `size` is less than 1024, then it will be multiplied by 1024. The value may also be specified with a k, K, m or M suffix. The number is then multiplied by 1000, 1024, 1000000, or 1048576, respectively. 4M (4194304) is the maximum allowed.

Unlike the `split-size-limit` option, this allows the recipient of the shell archives to be unpacked in any order.

split-size-limit option (-L).

This is the "split archive or files to *size*" option. This option takes an argument number `SIZE`.

This option has some usage constraints. It:

- is a member of the whole-size-limit class of options.
- must appear in combination with the following options: output-prefix.

Limit output file size to `size` bytes, splitting files if necessary. The allowed values are specified as with the `--whole-size-limit` option.

The archive parts created with this option must be unpacked in the correct order. If the recipient of the shell archives wants to put all of them in a single email folder (file), they will have to be saved in the correct order for unshar to unpack them all at once (using one of the split archive options). See Section 2.2 [unshar Invocation], page 16.

input-file-list option (-I).

This is the “read file list from a file” option. This option takes an argument string FILE. This option causes file to be reopened as standard input. If no files are found on the input line, then standard input is read for input file names. Use of this option will prohibit input files from being listed on the command line.

Input must be in a form similar to that generated by find, one filename per line. This switch is especially useful when the command line will not hold the list of files to be archived.

If the --intermix-type option is specified on the command line, then the compression options may be included in the standard input on lines by themselves and no file name may begin with a hyphen.

For example:

```
{ echo --compact xz
   find . -type f -print | sort
} | shar -S -p -L50K -o /somewhere/big
```

stdin-file-list option (-S).

This is the “read file list from standard input” option. This option is actually a no-op. It is a wrapper for --input-file-list=-.

NOTE: THIS OPTION IS DEPRECATED

2.1.5 headers options

Controlling the shar headers.

archive-name option (-n).

This is the “use name to document the archive” option. This option takes an argument string NAME. Name of archive to be included in the subject header of the shar files. See the --net-headers option.

submitter option (-s).

This is the “override the submitter name” option. This option takes an argument string WHO@WHERE. shar will normally determine the submitter name by querying the system. Use this option if it is being done on behalf of another.

net-headers option (-a).

This is the “output submitted-by: & archive-name: headers” option.

This option has some usage constraints. It:

- must appear in combination with the following options: archive-name.

Adds specialized email headers:

```
Submitted-by: who@where
Archive-name: name/part##
```

The *who@where* is normally derived, but can be specified with the `--submitter` option. The *name* must be provided with the `--archive-name` option. If the archive name includes a slash (`/`) character, then the `/part##` is omitted. Thus '`-n xyzzy`' produces:

```
xyzzy/part01
xyzzy/part02
```

while '`-n xyzzy/patch`' produces:

```
xyzzy/patch01
xyzzy/patch02
```

and '`-n xyzzy/patch01.`' produces:

```
xyzzy/patch01.01
xyzzy/patch01.02
```

cut-mark option (-c).

This is the "start the shar with a cut line" option. A line saying 'Cut here' is placed at the start of each output file.

translate option (-t).

This is the "translate messages in the script" option. Translate messages in the script. If you have set the '`LANG`' environment variable, messages printed by `shar` will be in the specified language. The produced script will still be emitted using messages in the lingua franca of the computer world: English. This option will cause the script messages to appear in the languages specified by the '`LANG`' environment variable set when the script is produced.

2.1.6 xmit-defenses options

Protecting against transmission issues:.

no-character-count option.

This is the "do not use 'wc -c' to check size" option. Do NOT check each file with 'wc -c' after unpack. The default is to check.

no-md5-digest option (-D).

This is the "do not use `md5sum` digest to verify" option. Do *not* use `md5sum` digest to verify the unpacked files. The default is to check.

force-prefix option (-F).

This is the "apply the prefix character on every line" option. Forces the prefix character to be prepended to every line, even if not required. This option may slightly increase the size of the archive, especially if `--uuencode` or a compression option is used.

here-delimiter option (-d).

This is the “use *delim* to delimit the files” option. This option takes an argument string `DELIM`. Use DELIM to delimit the files in the shar instead of SHAR_EOF. This is for those who want to personalize their shar files. The delimiter will always be prefixed and suffixed with underscores.

2.1.7 shar-flavors options

Producing different kinds of shars:.

vanilla-operation option (-V).

This is the “produce very simple shars” option. This option produces ‘`vanilla`’ shars which rely only upon the existence of `echo`, `test` and `sed` in the unpacking environment.

It changes the default behavior from mixed mode (`--mixed-uuencode`) to text mode (`--text-files`). Warnings are produced if options are specified that will require decompression or decoding in the unpacking environment.

no-piping option (-P).

This is the “use temporary files between programs” option. In the `shar` file, use a temporary file to hold file contents between unpacking stages instead of using pipes. This option is mandatory when you know the unpacking will happen on systems that do not support pipes.

no-check-existing option (-x).

This is the “blindly overwrite existing files” option. Create the archive so that when processed it will overwrite existing files without checking first. If neither this option nor the `--query-user` option is specified, the unpack will not overwrite pre-existing files. In all cases, however, if `--cut-mark` is passed as a parameter to the script when unpacking, then existing files will be overwritten unconditionally.

```
        sh shar-archive-file -c
```

query-user option (-X).

This is the “ask user before overwriting files” option.

This option has some usage constraints. It:

- must not appear in combination with any of the following options: vanilla-operation.

When unpacking, interactively ask the user if files should be overwritten. Do not use for shars submitted to the net.

Use of this option produces shars which *will* cause problems with some unshar-style procedures, particularily when used together with vanilla mode (`--vanilla-operation`). Use this feature mainly for archives to be passed among agreeable parties. Certainly, `-X` is *not* for shell archives which are to be submitted to Usenet or other public networks.

The problem is that `unshar` programs or procedures often feed `/bin/sh` from its standard input, thus putting `/bin/sh` and the shell archive script in competition for input lines. As an attempt to alleviate this problem, `shar` will try to detect if `/dev/tty` exists at the receiving site and will use it to read user replies. But this does not work in all cases, it may happen that the receiving user will have to avoid using `unshar` programs or procedures, and call `/bin/sh` directly. In vanilla mode, using `/dev/tty` is not even attempted.

no-timestamp option (-m).

This is the “do not restore modification times” option. Avoid generating 'touch' commands to restore the file modification dates when unpacking files from the archive.

When file modification times are not preserved, project build programs like "make" will see built files older than the files they get built from. This is why, when this option is not used, a special effort is made to restore timestamps.

quiet-unshar option (-Q).

This is the “avoid verbose messages at unshar time” option. Verbose OFF. Disables the inclusion of comments to be output when the archive is unpacked.

basename option (-f).

This is the “restore in one directory, despite hierarchy” option. Restore by the base file name only, rather than path. This option causes only file names to be used, which is useful when building a shar from several directories, or another directory. Note that if a directory name is passed to shar, the substructure of that directory will be restored whether this option is specified or not.

2.1.8 internationalization options

Internationalization options:.

no-i18n option.

This is the “do not internationalize” option. Do not produce internationalized shell archives, use default English messages. By default, shar produces archives that will try to output messages in the unpackers preferred language (as determined by the LANG/LC_MESSAGES environmental variables) when they are unpacked. If no message file for the unpackers language is found at unpack time, messages will be in English.

print-text-domain-dir option.

This is the “print directory with shar messages” option. Prints the directory shar looks in to find messages files for different languages, then immediately exits.

2.1.9 feedback options

user feedback/entertainment.

quiet option (-q).

This is the “do not output verbose messages” option. omit progress messages.

silent option.

This is an alias for the quiet option, see [shar quiet], page 13.

2.1.10 presetting/configuring shar

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("rc" or "ini") files.

libopts will search in $HOME for configuration (option) data. The environment variable HOME, is expanded and replaced when the program runs If this is a plain file, it is simply processed. If it is a directory, then a file named .sharrc is searched for within that directory.

Configuration files may be in a wide variety of formats. The basic format is an option name followed by a value (argument) on the same line. Values may be separated from the option name with a colon, equal sign or simply white space. Values may be continued across multiple lines by escaping the newline with a backslash.

Multiple programs may also share the same initialization file. Common options are collected at the top, followed by program specific segments. The segments are separated by lines like:

```
[SHAR]
```

or by

```
<?program shar>
```

Do not mix these styles within one configuration file.

Compound values and carefully constructed string values may also be specified using XML syntax:

```
<option-name>
   <sub-opt>...&lt;...&gt;...</sub-opt>
</option-name>
```

yielding an option-name.sub-opt string value of

```
"...<...>..."
```

AutoOpts does not track suboptions. You simply note that it is a hierarchicly valued option. AutoOpts does provide a means for searching the associated name/value pair list (see: optionFindValue).

The command line options relating to configuration and/or usage help are:

version (-v)

Print the program version to standard out, optionally with licensing information, then exit 0. The optional argument specifies how much licensing detail to provide. The default is to print the license name with the version. The licensing infomation may be selected with an option argument. Only the first letter of the argument is examined:

'version' Only print the version.

'copyright'
Name the copyright usage licensing terms. This is the default.

'verbose' Print the full copyright usage licensing terms.

2.1.11 shar exit status

One of the following exit values will be returned:

'0 (EXIT_SUCCESS)'
Successful program execution.

'1 (EXIT_OPTION_ERROR)'
The command options were misconfigured.

'2 (EXIT_FILE_NOT_FOUND)'
a specified input could not be found

'3 (EXIT_CANNOT_OPENDIR)'
open/close of specified directory failed

'4 (EXIT_FAILED)'
Resource limit/miscelleaneous shar command failure

'63 (EXIT_BUG)'
There is a shar command bug. Please report it.

'66 (EX_NOINPUT)'
A specified configuration file could not be loaded.

'70 (EX_SOFTWARE)'
libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

2.1.12 shar Authors

The `shar` and `unshar` programs is the collective work of many authors. Many people contributed by reporting problems, suggesting various improvements or submitting actual code. A list of these people is in the `THANKS` file in the sharutils distribution.

2.1.13 shar Bugs

Please put '`sharutils`' in the subject line for emailed bug reports. It helps to spot the message.

2.1.14 shar Examples

The first shows how to make a shell archive out of all C program sources. The second produces a shell archive with all `.c` and `.h` files, which unpacks silently. The third gives a shell archive of all uuencoded `.arc` files, into numbered files starting from `arc.sh.01`. The last example gives a shell archive which will use only the file names at unpack time.

```
shar *.c > cprog.shar
shar -Q *.[ch] > cprog.shar
shar -B -l28 -oarc.sh *.arc
shar -f /lcl/src/u*.c > u.sh
```

2.1.15 shar Warnings

No attempt is made to restore the protection and modification dates for directories, even if this is done by default for files. Thus, if a directory is given to `shar`, the protection and modification dates of corresponding unpacked directory may not match those of the original.

If a directory is passed to shar, it may be scanned more than once, to conserve memory. Therefore, do not change the directory contents while shar is running.

Be careful that the output file(s) are not included in the inputs or shar may loop until the disk fills up. Be particularly careful when a directory is passed to shar that the output files are not in that directory or a subdirectory of it.

Use of the compression and encoding options will slow the archive process, perhaps considerably.

Use of the `--query-user` produces shars which *will* cause problems with many unshar procedures. Use this feature only for archives to be passed among agreeable parties. Certainly, `query-user` is NOT for shell archives which are to be distributed across the net. The use of compression in net shars will cause you to be flamed off the earth. Not using the `--no-timestamp` or `--force-prefix` options may also get you occasional complaints. Put these options into your `~/.sharrc` file.

2.1.16 shar See Also

unshar(1)

2.2 Invoking unshar

Unshar scans the input files (typically email messages) looking for the start of a shell archive. If no files are given, then standard input is processed instead. It then passes each archive discovered through an invocation of the shell program to unpack it.

This section was generated by **AutoGen**, using the `agtexi-cmd` template and the option descriptions for the `unshar` program. This software is released under the GNU General Public License, version 3 or later.

2.2.1 unshar help/usage (`--help`)

This is the automatically generated usage text for unshar.

The text printed is the same whether selected with the `help` option (`--help`) or the `more-help` option (`--more-help`). `more-help` will print the usage text by passing it through a pager program. `more-help` is disabled on platforms without a working `fork(2)` function. The `PAGER` environment variable is used to select the program, defaulting to `more`. Both will exit with a status code of 0.

```
Usage: /u/ROOT/usr/local/bin/unshar [OPTION]... [FILE]...
Mandatory arguments to long options are mandatory for short options too.

  -d, --directory=DIRECTORY   change to DIRECTORY before unpacking
  -c, --overwrite             pass -c to shar script for overwriting files
  -e, --exit-0                same as ‘--split-at="exit 0"’
  -E, --split-at=STRING       split concatenated shars after STRING
  -f, --force                 same as ‘-c’
      --help                  display this help and exit
      --version               output version information and exit
```

```
If no FILE, standard input is read.
Report bugs to <bug-gnu-utils@gnu.org>.
```

2.2.2 directory option (-d)

This is the “change directory to `dir` first” option. This option takes an argument string `dir`. Before unpacking any files, change the current directory to `dir`.

2.2.3 overwrite option (-c)

This is the “overwrite any pre-existing files” option. This option is passed through as an option to the shar file. Many shell archive scripts accept a `-c` argument to indicate that existing files should be overwritten.

2.2.4 force option (-f)

This is an alias for the overwrite option, see Section 2.2.3 [unshar overwrite], page 17.

2.2.5 split-at option (-E)

This is the “separate archives on *split-pat* lines” option. This option takes an argument string `split-pat`. With this option, `unshar` isolates each different shell archive from the others which have been placed in the same file, unpacking each in turn, from the beginning of the file to the end. Its proper operation relies on the fact that many shar files are terminated by a readily identifiable string.

For example, noticing that most ‘.signatures’ have a double hyphen ("–") on a line right before them, one can then sometimes use `--split-at=--`. The signature will then be skipped, along with the headers of the following message.

2.2.6 exit-0 option (-e)

This is the “split archives at "exit 0" lines” option.

This option has some usage constraints. It:

- must not appear in combination with any of the following options: split-at.

Most shell archives end with a line consisting of simply "exit 0". This option is equivalent to (and conflicts with) `--split-at="exit 0"`.

2.2.7 presetting/configuring unshar

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("rc" or "ini") files.

`libopts` will search in `$HOME` for configuration (option) data. The environment variable `HOME,` is expanded and replaced when the program runs If this is a plain file, it is simply processed. If it is a directory, then a file named `.sharrc` is searched for within that directory.

Configuration files may be in a wide variety of formats. The basic format is an option name followed by a value (argument) on the same line. Values may be separated from the option name with a colon, equal sign or simply white space. Values may be continued across multiple lines by escaping the newline with a backslash.

Multiple programs may also share the same initialization file. Common options are collected at the top, followed by program specific segments. The segments are separated by lines like:

```
[UNSHAR]
```

or by

```
<?program unshar>
```

Do not mix these styles within one configuration file.

Compound values and carefully constructed string values may also be specified using XML syntax:

```
<option-name>
   <sub-opt>...&lt;...&gt;...</sub-opt>
</option-name>
```

yielding an `option-name.sub-opt` string value of

```
"...<...>..."
```

`AutoOpts` does not track suboptions. You simply note that it is a hierarchicly valued option. `AutoOpts` does provide a means for searching the associated name/value pair list (see: optionFindValue).

The command line options relating to configuration and/or usage help are:

version (-v)

Print the program version to standard out, optionally with licensing information, then exit 0. The optional argument specifies how much licensing detail to provide. The default is to print the license name with the version. The licensing infomation may be selected with an option argument. Only the first letter of the argument is examined:

‘`version`’ Only print the version.

‘`copyright`’
Name the copyright usage licensing terms. This is the default.

‘`verbose`’ Print the full copyright usage licensing terms.

2.2.8 unshar exit status

One of the following exit values will be returned:

‘`0 (EXIT_SUCCESS)`’
Successful program execution.

‘`1 (EXIT_FAILURE)`’
The operation failed or the command syntax was not valid.

‘`66 (EX_NOINPUT)`’
A specified configuration file could not be loaded.

‘`70 (EX_SOFTWARE)`’
libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

2.2.9 unshar Authors

The shar and unshar programs is the collective work of many authors. Many people contributed by reporting problems, suggesting various improvements or submitting actual code. A list of these people is in the THANKS file in the sharutils distribution.

2.2.10 unshar Bugs

Please put 'sharutils' in the subject line for emailed bug reports. It helps to spot the message.

2.2.11 unshar See Also

shar(1)

2.3 Invoking uuencode

uuencode is used to create an ASCII representation of a file that can be sent over channels that may otherwise corrupt the data. Specifically, email cannot handle binary data and will often even insert a character when the six character sequence "\nFrom " is seen.

uuencode will read in-file if provided and otherwise read data from standard in and write the encoded form to standard out. The output will begin with a header line for use by uudecode giving it the resulting suggested file output-name and access mode. If the output-name is specifically /dev/stdout, then uudecode will emit the decoded file to standard out.

Note: uuencode uses buffered input and assumes that it is not hand typed from a tty. The consequence is that at a tty, you may need to hit Ctl-D several times to terminate input.

This section was generated by **AutoGen**, using the agtexi-cmd template and the option descriptions for the uuencode program. This software is released under the GNU General Public License, version 3 or later.

2.3.1 uuencode help/usage (--help)

This is the automatically generated usage text for uuencode.

The text printed is the same whether selected with the help option (--help) or the more-help option (--more-help). more-help will print the usage text by passing it through a pager program. more-help is disabled on platforms without a working fork(2) function. The PAGER environment variable is used to select the program, defaulting to more. Both will exit with a status code of 0.

```
uuencode (GNU sharutils) - encode a file into email friendly text - Ver. 4.13.3
USAGE:  uuencode [ -<flag> | --<name> ]... [ in-file ] output-name

   -m, --base64               convert using base 64
   -e, --encode-file-name     encode the output file name
   -v, --version[=arg]        Output version information and exit
   -h, --help                 Display extended usage information and exit
   -!, --more-help            Extended usage information passed thru pager
   -R, --save-opts[=arg]      Save the option state to a config file
   -r, --load-opts=str        Load options from a config file
                                - disabled as --no-load-opts
                                - may appear multiple timess

Options are specified by doubled hyphens and their name or by a single
hyphen and the flag character.

The following option preset mechanisms are supported:
 - reading file $HOME/.sharrc

``uuencode'' is used to create an ASCII representation of a file that can
be sent over channels that may otherwise corrupt the data.  Specifically,
email cannot handle binary data and will often even insert a character when
the six character sequence "\nFrom " is seen.

``uuencode'' will read ``in-file'' if provided and otherwise read data from
standard in and write the encoded form to standard out.  The output will
begin with a header line for use by ``uudecode'' giving it the resulting
suggested file ``output-name'' and access mode.  If the ``output-name'' is
specifically ``/dev/stdout'', then ``uudecode'' will emit the decoded file
to standard out.

``Note'': ``uuencode'' uses buffered input and assumes that it is not hand
typed from a tty.  The consequence is that at a tty, you may need to hit
Ctl-D several times to terminate input.

please send bug reports to:  bug-gnu-utils@gnu.org
```

2.3.2 base64 option (-m)

This is the “convert using base 64” option. By default, `uuencode` will encode using the traditional conversion. It is slower and less compact than base 64. The encoded form of the file is expanded by 37% for UU encoding and by 35% for base64 encoding (3 bytes become 4 plus control information).

2.3.3 encode-file-name option (-e)

This is the “encode the output file name” option. Since output file names may contain characters that are not handled well by various transmission modes, you may specify that the `output-name` be hex encoded as well.

2.3.4 presetting/configuring uuencode

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("rc" or "ini") files.

`libopts` will search in `$HOME` for configuration (option) data. The environment variable `HOME,` is expanded and replaced when the program runs If this is a plain file, it is simply processed. If it is a directory, then a file named `.sharrc` is searched for within that directory.

Configuration files may be in a wide variety of formats. The basic format is an option name followed by a value (argument) on the same line. Values may be separated from the option name with a colon, equal sign or simply white space. Values may be continued across multiple lines by escaping the newline with a backslash.

Multiple programs may also share the same initialization file. Common options are collected at the top, followed by program specific segments. The segments are separated by lines like:

```
[UUENCODE]
```

or by

```
<?program uuencode>
```

Do not mix these styles within one configuration file.

Compound values and carefully constructed string values may also be specified using XML syntax:

```
<option-name>
   <sub-opt>...&lt;...&gt;...</sub-opt>
</option-name>
```

yielding an `option-name.sub-opt` string value of

```
"...<...>..."
```

`AutoOpts` does not track suboptions. You simply note that it is a hierarchicly valued option. `AutoOpts` does provide a means for searching the associated name/value pair list (see: optionFindValue).

The command line options relating to configuration and/or usage help are:

version (-v)

Print the program version to standard out, optionally with licensing information, then exit 0. The optional argument specifies how much licensing detail to provide. The default is to print the license name with the version. The licensing infomation may be selected with an option argument. Only the first letter of the argument is examined:

'`version`' Only print the version.

'`copyright`'
Name the copyright usage licensing terms. This is the default.

'`verbose`' Print the full copyright usage licensing terms.

2.3.5 uuencode exit status

One of the following exit values will be returned:

'`0 (EXIT_SUCCESS)`'
Successful program execution.

'`1 (EXIT_FAILURE)`'
The operation failed or the command syntax was not valid.

'`66 (EX_NOINPUT)`'
A specified configuration file could not be loaded.

'70 (EX_SOFTWARE)'
: libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

2.3.6 uuencode Bugs

Please put 'sharutils' in the subject line for emailed bug reports. It helps to spot the message.

The output file name must not begin with the 11 character sequence:

```
hex-encode:
```

unless you specify the -h (encode-file-name) option.

2.3.7 uuencode Standards

This implementation is compliant with P1003.2b/D11.

2.3.8 uuencode History

The uuencode command first appeared in BSD 4.0.

2.3.9 uuencode See Also

uudecode(1)

2.4 Invoking uudecode

If no file(s) are provided, then standard input is decoded. Uudecode transforms uuencoded files into their original form.

The encoded file(s) may be specified on the command line, or one may be read from standard input. The output file name is specified in the encoded file, but may be overridden with the -o option. It will have the mode of the original file, except that setuid and execute bits are not retained. If the output file is specified to be /dev/stdout or -, the result will be written to standard output. If there are multiple input files and the second or subsquent file specifies standard output, the decoded data will be written to the same file as the previous output. Don't do that.

uudecode ignores any leading and trailing lines. It looks for a line that starts with "'begin'" and proceeds until the end-of-encoding marker is found. The program determines from the header line of the encoded file which of the two supported encoding schemes was used.

This section was generated by **AutoGen**, using the agtexi-cmd template and the option descriptions for the uudecode program. This software is released under the GNU General Public License, version 3 or later.

2.4.1 uudecode help/usage (--help)

This is the automatically generated usage text for uudecode.

The text printed is the same whether selected with the help option (--help) or the more-help option (--more-help). more-help will print the usage text by passing it through a pager program. more-help is disabled on platforms without a working fork(2) function.

The PAGER environment variable is used to select the program, defaulting to more. Both will exit with a status code of 0.

```
uudecode (GNU sharutils) - decode an encoded file - Ver. 4.13.3
USAGE:  uudecode [ -<flag> [<val>] | --<name>[{=| }<val>] ]... [ file ... ]

   -o, --output-file=str      direct output to file
   -c, --ignore-chmod         Ignore fchmod(3P) errors
   -v, --version[=arg]        Output version information and exit
   -h, --help                 Display extended usage information and exit
   -!, --more-help            Extended usage information passed thru pager
   -R, --save-opts[=arg]      Save the option state to a config file
   -r, --load-opts=str        Load options from a config file
                                - disabled as --no-load-opts
                                - may appear multiple timess

Options are specified by doubled hyphens and their name or by a single
hyphen and the flag character.

If no ‘‘file’’(s) are provided, then standard input is decoded.

The following option preset mechanisms are supported:
 - reading file $HOME/.sharrc

‘‘Uudecode’’ transforms uuencoded files into their original form.

The encoded file(s) may be specified on the command line, or one may be
read from standard input.  The output file name is specified in the encoded
file, but may be overridden with the ‘‘-o’’ option.  It will have the mode
of the original file, except that setuid and execute bits are not retained.
If the output file is specified to be ‘‘/dev/stdout’’ or ‘‘-’’, the result
will be written to standard output.  If there are multiple input files and
the second or subsquent file specifies standard output, the decoded data
will be written to the same file as the previous output.  Don’t do that.

‘‘uudecode’’ ignores any leading and trailing lines.  It looks for a line
that starts with "‘‘begin’’" and proceeds until the end-of-encoding marker
is found.  The program determines from the header line of the encoded file
which of the two supported encoding schemes was used.

please send bug reports to:  bug-gnu-utils@gnu.org
```

2.4.2 output-file option (-o)

This is the “direct output to file” option. This option takes an argument string file. If specifed, decoded data are written to this file. When multiple inputs are specified on the command line, this option cannot be specified. All decoded data must be written to the file name encoded in the data.

2.4.3 ignore-chmod option (-c)

This is the “ignore `fchmod(3p)` errors” option. By default, if the output file permissions cannot be changed to the permissions specified in the encoded data, the file will not be written out and execution stops. This option will cause that error to be ignored. The resulting file will have all the data, but the incorrect mode settings.

`fchmod()` errors are also ignored if `POSIXLY_CORRECT` is set in the environment. RE: ‘`http://austingroupbugs.net/view.php?id=635`’

A warning is always emitted when `fchmod()` fails.

2.4.4 presetting/configuring uudecode

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("rc" or "ini") files.

`libopts` will search in `$HOME` for configuration (option) data. The environment variable `HOME,` is expanded and replaced when the program runs If this is a plain file, it is simply processed. If it is a directory, then a file named `.sharrc` is searched for within that directory.

Configuration files may be in a wide variety of formats. The basic format is an option name followed by a value (argument) on the same line. Values may be separated from the option name with a colon, equal sign or simply white space. Values may be continued across multiple lines by escaping the newline with a backslash.

Multiple programs may also share the same initialization file. Common options are collected at the top, followed by program specific segments. The segments are separated by lines like:

```
[UUDECODE]
```

or by

```
<?program uudecode>
```

Do not mix these styles within one configuration file.

Compound values and carefully constructed string values may also be specified using XML syntax:

```
<option-name>
   <sub-opt>...&lt;...&gt;...</sub-opt>
</option-name>
```

yielding an `option-name.sub-opt` string value of

```
"...<...>..."
```

`AutoOpts` does not track suboptions. You simply note that it is a hierarchicly valued option. `AutoOpts` does provide a means for searching the associated name/value pair list (see: optionFindValue).

The command line options relating to configuration and/or usage help are:

version (-v)

Print the program version to standard out, optionally with licensing information, then exit 0. The optional argument specifies how much licensing detail to provide. The default is to print the license name with the version. The licensing infomation may be selected with an option argument. Only the first letter of the argument is examined:

'version' Only print the version.

'copyright'
Name the copyright usage licensing terms. This is the default.

'verbose' Print the full copyright usage licensing terms.

2.4.5 uudecode exit status

One of the following exit values will be returned:

'0 (EXIT_SUCCESS)'
Successful program execution.

'1 (EXIT_OPTION_ERROR)'
The command options were misconfigured.

'2 (EXIT_INVALID)'
(warning) One or more input files contained no valid data

'4 (EXIT_NO_INPUT)'
(warning) The specified input file was not found

'8 (EXIT_NO_OUTPUT)'
The specified output file could not be created (error); or else one of the output files could not be written or its access mode could not be changed (warnings). The accompanying message(s) will distinguish.

'66 (EX_NOINPUT)'
A specified configuration file could not be loaded.

'70 (EX_SOFTWARE)'
libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

2.4.6 uudecode Bugs

Please put 'sharutils' in the subject line for emailed bug reports. It helps to spot the message.

If more than one name in the encoded files are the same, or if the second or following input files specifies standard output for the output file, then the result is probably not what is expected. Specifically, standard output will be appended to and named output files will be replaced.

2.4.7 uudecode Standards

This implementation is compliant with P1003.2b/D11.

2.4.8 uudecode See Also

uuencode(1)

Appendix A GNU Free Documentation License

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.