Linux Ubuntu 22.4.5 Manual Pages on command 'shar.1'

$ man shar.1

shar(1)   User Commands   shar(1)

NAME

    shar - create a shell archive

SYNOPSIS

    shar [-flags] [-flag [value]] [--option-name[[=|]value]] [<file>...]

    If no files are specified, the list of input files is read from standard input.

    Standard input must not be a terminal.

DESCRIPTION

    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 spec?

    ifying 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, base64 is

    the preferred encoding method.

OPTIONS

    Specifying compression

-p, --intermix-type

   specify compression for input files. This option must not appear in combi‐

   nation 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.

-C program, --compactor=program

   specify compaction (compression) program. This option may appear an unlim‐

   ited number of times. This option 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 pro‐

   gram name as the option name, e.g. --gzip. Those options, however, are be‐

   ing deprecated. There is also the xz compactor now. Specify xz with -C xz

   or --compactor=xz.

      Specifying the compactor "none" will disable file compression. Com‐

   pressed files are never processed as plain text. They are always uuencoded

   and the recipient must have uudecode to unpack them.

   Specifying the compactor compress is deprecated.

-g level, --level-of-compression=level

   pass LEVEL for compression. This option takes an integer number as its ar‐

   gument. The value of level is constrained to being:

     in the range 1 through 9

   The default level for this option is:

      9

   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.

-j, --bzip2

   bzip2 and uuencode files. This option 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

-z, --gzip

    gzip and uuencode files. This option 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

-Z, --compress

    compress and uuencode files. This option may appear an unlimited number of

    times.

    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

    This is an alias for the --level-of-compression option.

    NOTE: THIS OPTION IS DEPRECATED

-b bits, --bits-per-code=bits

    pass bits (default 12) to compress. The default bits for this option is:

        12

    This is the compression factor used by the compress program.

    NOTE: THIS OPTION IS DEPRECATED

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 process?

    ing.

    -M, --mixed-uuencode

        decide uuencoding for each file. This option is a member of the mixed-uuen?

        code class of options.

        Automatically determine if the files are text or binary and archive cor?

        rectly. 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:

The file does not contain any ASCII control character besides BS
(backspace), HT (horizontal tab), LF (new line) or FF (form feed).

The file contains no character with its eighth-bit set.

The file contains no line beginning with the five letters "from ", capital?
ized or not. (Mail handling programs will often gratuitously insert a >
character before it.)

The file is either empty or ends with a LF (newline) byte.

No line in the file contains more than 200 characters. For counting pur?
pose, lines are separated by a LF (newline).

-B, --uuencode

treat all files as binary. This option is a member of the mixed-uuencode
class of options.

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

-T, --text-files

treat all files as text. This option is a member of the mixed-uuencode
class of options.

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

Specifying file selection and output modes

-o prefix, --output-prefix=prefix

print output to file PREFIX.nn.

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 pre?
fix does not contain such a % character, the string .%02d is internally ap?
pended.

-l size, --whole-size-limit=size

split archive, not files, to size. This option is a member of the whole-

size-limit class of options. This option must appear in combination with the following options: output-prefix. This option takes an integer number as its argument. The value of size is constrained to being:

in the range 8 through 1023, or

in the range 8192 through 4194304

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 maxi?mum allowed.

Unlike the split-size-limit option, this allows the recipient of the shar files to unpack them in any order.

-L size, --split-size-limit=size

split archive or files to size. This option is a member of the whole-size-limit class of options. This option must appear in combination with the following options: output-prefix. This option takes an integer number as its argument. The value of size is constrained to being:

in the range 8 through 1023, or

in the range 8192 through 4194304

Limit output file size to size bytes, splitting files if necessary. The al?lowed 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 op?tions). see: unshar Invocation.

-I file, --input-file-list=file

read file list from a 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 com?mand 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 them?

selves 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

  -S, --stdin-file-list

    read file list from standard input.

    This option is actually a no-op. It is a wrapper for --input-file-list=-.

    NOTE: THIS OPTION IS DEPRECATED

Controlling the shar headers

  -n name, --archive-name=name

    use name to document the archive.

    Name of archive to be included in the subject header of the shar files. See

    the --net-headers option.

  -s who@where, --submitter=who@where

    override the submitter name.

    shar will normally determine the submitter name by querying the system. Use

    this option if it is being done on behalf of another.

  -a, --net-headers

    output Submitted-by: & Archive-name: headers. This option 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 --submit?

    ter option. The name must be provided with the --archive-name option. If

    the archive name includes a slash (/) character, then the /part## is omit?

    ted. 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

-c, --cut-mark

start the shar with a cut line.

A line saying 'Cut here' is placed at the start of each output file.

-t, --translate

translate messages in the script.

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.

Protecting against transmission issues

--no-character-count

do not use `wc -c' to check size.

Do NOT check each file with 'wc -c' after unpack. The default is to check.

-D, --no-md5-digest

do not use md5sum digest to verify.

Do not use md5sum digest to verify the unpacked files. The default is to

check.

-F, --force-prefix

apply the prefix character on every line.

Forces the prefix character to be prepended to every line, even if not re?

quired. This option may slightly increase the size of the archive, espe?

cially if --uuencode or a compression option is used.

-d delim, --here-delimiter=delim

use delim to delimit the files. The default delim for this option is:

SHAR_EOF

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.

Producing different kinds of shars

-V, --vanilla-operation

produce very simple shars.

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.

-P, --no-piping

use temporary files between programs.

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

-x, --no-check-existing

blindly overwrite existing files.

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

-X, --query-user

ask user before overwriting files. This option must not appear in combina‐
tion 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 un‐
shar-style procedures, particularly 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 proce? dures, and call /bin/sh directly. In vanilla mode, using /dev/tty is not even attempted.

-m, --no-timestamp

do not restore modification times.

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.

-Q, --quiet-unshar

avoid verbose messages at unshar time.

Verbose OFF. Disables the inclusion of comments to be output when the ar? chive is unpacked.

-f, --basename

restore in one directory, despite hierarchy.

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 sev? eral 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.

Internationalization options

--no-i18n

do not internationalize.

Do not produce internationalized shell archives, use default English mes?

sages. 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

print directory with shar messages.

Prints the directory shar looks in to find messages files for different lan?

guages, then immediately exits.

User feedback/entertainment

-q, --quiet

do not output verbose messages.

omit progress messages.

--silent

This is an alias for the --quiet option.

-h, --help

Display usage information and exit.

-!, --more-help

Pass the extended usage information through a pager.

-R [cfgfile], --save-opts [=cfgfile]

Save the option state to cfgfile. The default is the last configuration

file listed in the OPTION PRESETS section, below. The command will exit af?

ter updating the config file.

-r cfgfile, --load-opts=cfgfile, --no-load-opts

Load options from cfgfile. The no-load-opts form will disable the loading

of earlier config/rc/ini files. --no-load-opts is handled early, out of or?

der.

-v [{v|c|n --version [{v|c|n}]}]

Output version of program and exit. The default mode is `v', a simple ver?

sion. The `c' mode will print copyright information and `n' will print the

full copyright notice.

OPTION PRESETS

Any option that is not marked as not presettable may be preset by loading values

from configuration ("RC" or ".INI") file(s). The file "$HOME/.sharrc" will be

used, if present.

WARNINGS

No attempt is made to restore the protection and modification dates for directo?

ries, 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 par?

ties. 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.

FILES

See OPTION PRESETS for configuration files.

EXAMPLES

The first shows how to make a shell archive out of all C program sources. The sec?

ond 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 start?

ing 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

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.

SEE ALSO

   unshar(1)

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 distri?

   bution.

COPYRIGHT

   Copyright (C) 1994-2015 Free Software Foundation, Inc. all rights reserved. This

   program is released under the terms of the GNU General Public License, version 3 or

   later.

BUGS

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

   the message.

   Please send bug reports to: bug-gnu-utils@gnu.org

NOTES

This manual page was AutoGen-erated from the shar option definitions.