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

$ man unshar.1

unshar(1)   User Commands   unshar(1)

NAME

   unshar - unpack a shar archive

SYNOPSIS

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

   The operands that this program operates on may be specified either on the command

   line or read from standard input, one per line. In that input, leading and trail?

   ing white space is stripped, blank lines are ignored. Standard input may not be a

   terminal.

DESCRIPTION

   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 program will perform its function for every file named on the command line or

   every file named in a list read from stdin. The arguments or input names must be

   pre-existing files. The input list may contain comments, which are blank lines or

   lines beginning with a '#' character.

OPTIONS

   -d dir, --directory=dir

        unpack into the directory dir.

        The input file names are relative to the current directory when the program

was started. This option tells unshar to insert a cd <dir> commad at the start of the shar text written to the shell.

-c, --overwrite

overwrite any pre-existing files.

This option is passed through as an option to the shar file. Many shell ar‐ chive scripts accept a -c argument to indicate that existing files should be overwritten.

-f, --force

This is an alias for the --overwrite option.

-E split-mark, --split-at=split-mark

split input on split-mark lines. The default split-mark for this option is:

    exit 0

With this option, unshar isolates each different shell archive from the oth‐ ers 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 at the start of the last line.

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 mes‐ sage.

-e, --exit-0

split input on "exit 0" lines. This option 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 op‐ tion is equivalent to (and conflicts with) --split-at="exit 0".

-D, --debug

debug the shell code.

"set -x" will be emitted into the code the shell interprets.

-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.

FILES

See OPTION PRESETS for configuration files.

EXIT STATUS

One of the following exit values will be returned:

0 (EXIT_SUCCESS)

>   Successful program execution.

1 (EXIT_FAILURE)

>   There was an error in command usage.

2 (EXIT_POPEN_PROBLEM)

>   cannot spawn or write to a shell process

3 (EXIT_CANNOT_CREATE)

>   cannot create output file

4 (EXIT_BAD_DIRECTORY)

>   the working directory structure is invalid

5 (EXIT_NOMEM)

>   memory allocation failure

6 (EXIT_INVALID)

invalid input, does not contain a shar file

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

shar(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 unshar option definitions.

GNU sharutils (4.15.2)   30 May 2015   unshar(1)