

GNU CPIO

2.14 28 April 2023

by Robert Carleton and Sergey Poznyakoff

This manual documents GNU cpio (version 2.14, 28 April 2023).

Copyright © 1995–2023 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover texts being “A GNU Manual”, and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled “GNU Free Documentation License”.

(a) The FSF’s Back-Cover Text is: “You have freedom to copy and modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development.”

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

1 Introduction

GNU `cpio` copies files into or out of a `cpio` or `tar` archive. The archive can be another file on the disk, a magnetic tape, or a pipe.

GNU `cpio` supports the following archive formats: binary, old ASCII, new ASCII, `crc`, HPUX binary, HPUX old ASCII, old `tar`, and POSIX.1 `tar`. The `tar` format is provided for compatibility with the `tar` program. By default, `cpio` creates binary format archives, for compatibility with older `cpio` programs. When extracting from archives, `cpio` automatically recognizes which kind of archive it is reading and can read archives created on machines with a different byte-order.

2 Tutorial

GNU `cpio` performs three primary functions. Copying files to an archive, Extracting files from an archive, and passing files to another directory tree. An archive can be a file on disk, one or more floppy disks, or one or more tapes.

When creating an archive, `cpio` takes the list of files to be processed from the standard input, and then sends the archive to the standard output, or to the device defined by the `-F` option. See [Section 3.1 \[Copy-out mode\], page 2](#). Usually `find` or `ls` is used to provide this list to the standard input. In the following example you can see the possibilities for archiving the contents of a single directory.

```
% ls | cpio -ov > directory.cpio
```

The `-o` option creates the archive, and the `-v` option prints the names of the files archived as they are added. Notice that the options can be put together after a single `-` or can be placed separately on the command line. The `>` redirects the `cpio` output to the file `directory.cpio`.

If you wanted to archive an entire directory tree, the `find` command can provide the file list to `cpio`:

```
% find . -print -depth | cpio -ov > tree.cpio
```

This will take all the files in the current directory, the directories below and place them in the archive `tree.cpio`. Again the `-o` creates an archive, and the `-v` option shows you the name of the files as they are archived. See [Section 3.1 \[Copy-out mode\], page 2](#). Using the `.` in the `find` statement will give you more flexibility when doing restores, as it will save file names with a relative path vice a hard wired, absolute path. The `-depth` option forces `find` to print of the entries in a directory before printing the directory itself. This limits the effects of restrictive directory permissions by printing the directory entries in a directory before the directory name itself.

Extracting an archive requires a bit more thought. First of all, by default `cpio` extracts the files with exactly the same name as stored in the archive. That means that if the archive contains absolute paths, you will extract files to their absolute locations no matter what

directory you're in when running the command. You can instruct `cpio` to remove leading slashes using the `'--no-absolute-filenames'` option. Nevertheless, the good practice is to always test the archive using `cpio -t` prior to extracting it.

Furthermore, `cpio` will not create directories by default. Another characteristic, is it will not overwrite existing files unless you tell it to.

```
% cpio -iv < directory.cpio
```

This will retrieve the files archived in the file `'directory.cpio'` and restore them to their locations. The `'-i'` option extracts the archive and the `'-v'` shows the file names as they are extracted. If you are dealing with an archived directory tree, you need to use the `'-d'` option to create directories as necessary, something like:

```
% cpio -idv < tree.cpio
```

This will take the contents of the archive `'tree.cpio'` and extract it. If you try to extract the files on top of files of the same name that already exist (and have the same or later modification time) `cpio` will not extract the file unless told to do so by the `'-u'` option. See [Section 3.2 \[Copy-in mode\], page 4](#).

In copy-pass mode, `cpio` copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument. See [Section 3.3 \[Copy-pass mode\], page 6](#).

```
% find . -depth -print0 | cpio --null -pvd new-dir
```

The example shows copying the files of the present directory, and sub-directories to a new directory called `new-dir`. Some new options are the `'-print0'` available with GNU `find`, combined with the `'--null'` option of `cpio`. These two options act together to send file names between `find` and `cpio`, even if special characters are embedded in the file names. Another is `'-p'`, which tells `cpio` to pass the files it finds to the directory `'new-dir'`.

3 Invoking `cpio`

3.1 Copy-out mode

In copy-out mode, `cpio` copies files into an archive. It reads a list of filenames, one per line, on the standard input, and writes the archive onto the standard output. A typical way to generate the list of filenames is with the `find` command; you should give `find` the `'-depth'` option to minimize problems with permissions on directories that are unreadable.

Copy-out mode is requested by the `'-o'` (`'--create'`) command line option, e.g.:

```
% find | cpio -o > directory.cpio
```

The following options can be used in copy-out mode:

- '-0'
- '--null' Filenames in the list are delimited by ASCII null characters instead of newlines.
- '-A'
- '--append'
 - Append to an existing archive.
- '-a'
- '--reset-access-time'
 - Reset the access times of files after reading them.
- '--absolute-filenames'
 - Do not strip file system prefix components from the file names. This is the default.
- '--no-absolute-filenames'
 - Strip file system prefix components from the file names before storing them to the archive.
- '--block-size=*block-size*'
 - Sets the I/O block size to *block-size* * 512 bytes.
- '-B' Set the I/O block size to 5120 bytes.
- '-c' Use the old portable (ASCII) archive format.
- '-C *number*'
- '--io-size=*number*'
 - Set the I/O block size to the given *number* of bytes.
- '-D *dir*'
- '--directory=*dir*'
 - Change to directory *dir*
- '--force-local'
 - Treat the archive file as local, even if its name contains colons.
- '-F [[*user*@]*host*:]*archive-file*'
- '-O [[*user*@]*host*:]*archive-file*'
- '--file=[[*user*@]*host*:]*archive-file*'
 - Use the supplied *archive-file* instead of standard input. Optional *user* and *host* specify the user and host names in case of a remote archive.
- '-H *format*'
- '--format=*format*'
 - Use given archive format. See [\[format\]](#), page 11, for a list of available formats.
- '-L'
- '--dereference'
 - Dereference symbolic links (copy the files that they point to instead of copying the links).

`-M string`
`--message=string`
 Print *string* when the end of a volume of the backup media is reached.

`--quiet` Do not print the number of blocks copied.

`--rsh-command=command`
 Use *command* instead of `rsh` to access remote archives.

`-R`
`--owner=[user][:.] [group]`
 Set the ownership of all files created to the specified *user* and/or *group*. See [owner], page 13.

`-v`
`--verbose`
 Verbosely list the files processed.

`-V`
`--dot` Print a `‘.’` for each file processed.

`-W`
`--warning=flag`
 Control warning display. Argument is one of `‘none’`, `‘truncate’`, `‘no-truncate’` or `‘all’`. See [warning], page 14, for a detailed discussion of these.

3.2 Copy-in mode

In copy-in mode, `cpio` copies files from an archive into the filesystem or lists the archive contents. It reads the archive from the standard input. Any non-option command line arguments are shell globbing patterns; only files in the archive whose names match one or more of those patterns are copied from the archive. Unlike in the shell, an initial `‘.’` in a filename does match a wildcard at the start of a pattern, and a `‘/’` in a filename can match wildcards. If no patterns are given, all files are extracted.

The copy-in mode is requested by the `‘-i’` (`‘--extract’`) command line option.

The following options can be used in copy-in mode:

`--absolute-filenames`
 Do not strip file system prefix components from the file names. This is the default.

`--no-absolute-filenames`
 Create all files relative to the current directory.

`--block-size=block-size`
 Sets the I/O block size to *block-size* * 512 bytes.

`-b`
`--swap` Swap both halfwords of words and bytes of halfwords in the data. Equivalent to `‘-sS’`.

`-B` Set the I/O block size to 5120 bytes.

`-c` Use the old portable (ASCII) archive format.

`-C number`
`--io-size=number`
Set the I/O block size to the given *number* of bytes.

`-D dir`
`--directory=dir`
Change to directory *dir*

`-d`
`--make-directories`
Create leading directories where needed.

`-E file`
`--pattern-file=file`
Read additional patterns specifying filenames to extract or list from *file*.

`-f`
`--nonmatching`
Only copy files that do not match any of the given patterns.

`--force-local`
Treat the archive file as local, even if its name contains colons.

`-F [[user@]host:]archive-file`
`-I [[user@]host:]archive-file`
`--file=[[user@]host:]archive-file`
Use the supplied *archive-file* instead of standard input. Optional *user* and *host* specify the user and host names in case of a remote archive.

`-H format`
`--format=format`
Use given archive format. See [\[format\]](#), page 11, for a list of available formats.

`-m`
`--preserve-modification-time`
Retain previous file modification times when creating files.

`-M string`
`--message=string`
Print *string* when the end of a volume of the backup media is reached.

`--no-preserve-owner`
Do not change the ownership of the files.

`-n`
`--numeric-uid-gid`
In the verbose table of contents listing, show numeric UID and GID values.

`--only-verify-crc`
When reading a CRC format archive, only verify the CRC's of each file in the archive, don't actually extract the files

`--quiet` Do not print the number of blocks copied.

```

'--rsh-command=command'
    Use command instead of rsh to access remote archives.

'-r'
'--rename'
    Interactively rename files

'--sparse'
    Write files with large blocks of zeros as sparse files.

'-s'
'--swap-bytes'
    Swap the bytes of each halfword in the files

'-S'
'--swap-halfwords'
    Swap the halfwords of each word (4 bytes) in the files

'--to-stdout'
    Extract files to standard output.

'-u'
'--unconditional'
    Replace all files unconditionally.

'-v'
'--verbose'
    Verbosely list the files processed.

'-V'
'--dot'
    Print a '.' for each file processed.

'-W'

'--warning=flag'
    Control warning display. Argument is one of 'none', 'truncate', 'no-truncate'
    or 'all'. See [warning], page 14, for a detailed discussion of these.

```

3.3 Copy-pass mode

In copy-pass mode, `cpio` copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument.

This mode is requested by the '-p' ('--pass-through') command line option.

The following options are valid in copy-out mode:

```

'-0'
'--null'
    Filenames in the list are delimited by ASCII null characters instead of newlines.

'-a'
'--reset-access-time'
    Reset the access times of files after reading them.

```


`-b`
`--swap` Swap both halfwords of words and bytes of halfwords in the data. Equivalent to `-sS`.

`--block-size=block-size`
Sets the I/O block size to *block-size* * 512 bytes.

`-B` Set the I/O block size to 5120 bytes.

`-c` Use the old portable (ASCII) archive format.

`-C number`
`--io-size=number`
Set the I/O block size to the given *number* of bytes.

`-d`
`--make-directories`
Create leading directories where needed.

`--device-independent`
`--reproducible`
Create reproducible archives. This is equivalent to `--ignore-devno --ignore-dirnlink --renumber-inodes`.

`-D dir`
`--directory=dir`
Change to directory *dir*

`-E file`
`--pattern-file=file`
Read additional patterns specifying filenames to extract or list from *file*.

`-f`
`--nonmatching`
Only copy files that do not match any of the given patterns.

`-F [[user@]host:]archive-file`
`-O [[user@]host:]archive-file`
`--file=[[user@]host:]archive-file`
Use the supplied *archive-file* instead of standard input. Optional *user* and *host* specify the user and host names in case of a remote archive.

`--force-local`
Treat the archive file as local, even if its name contains colons.

`-H format`
`--format=format`
Use given archive format. See [\[format\]](#), page 11, for a list of available formats.

`--ignore-devno`
Store 0 in the device number field of each archive member, instead of the actual device number.

`--ignore-dirnlink`
Store 2 in the *nlink* field of each directory archive member, instead of the actual number of links.

'-l'
'--link' Link files instead of copying them, when possible.

'-L'
'--dereference'
Dereference symbolic links (copy the files that they point to instead of copying the links).

'-m'
'--preserve-modification-time'
Retain previous file modification times when creating files.

'-M *string*'
'--message=*string*'
Print *string* when the end of a volume of the backup media is reached.

'-n'
'--numeric-uid-gid'
In the verbose table of contents listing, show numeric UID and GID values.

'--no-preserve-owner'
Do not change the ownership of the files.

'--only-verify-crc'
When reading a CRC format archive, only verify the CRC's of each file in the archive, don't actually extract the files

'--quiet' Do not print the number of blocks copied.

'--rsh-command=*command*'
Use *command* instead of *rsh* to access remote archives.

'-r'
'--rename'
Interactively rename files

'--renumber-inodes'
Renumber inodes when storing them in the archive.

'-R'
'--owner=[*user*][:.] [*group*]'
Set the ownership of all files created to the specified *user* and/or *group*. See [owner], page 13.

'-s'
'--swap-bytes'
Swap the bytes of each halfword in the files

'--sparse'
Write files with large blocks of zeros as sparse files.

'-S'
'--swap-halfwords'
Swap the halfwords of each word (4 bytes) in the files

`--to-stdout`
Extract files to standard output.

`-u`
`--unconditional`
Replace all files unconditionally.

`-v`
`--verbose`
Verbosely list the files processed.

`-V`
`--dot` Print a `'.'` for each file processed.

`-W`
`--warning=flag`
Control warning display. Argument is one of `'none'`, `'truncate'`, `'no-truncate'` or `'all'`. See [warning], page 14, for a detailed discussion of these.

3.4 Options

This section summarizes all available command line options. References in square brackets after each option indicate `cpio` modes in which this option is valid.

`-0`
`--null` [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Read a list of filenames terminated by a null character, instead of a newline, so that files whose names contain newlines can be archived. GNU `find` is one way to produce a list of null-terminated filenames. This option may be used in copy-out and copy-pass modes.

`-a`
`--reset-access-time` [[copy-out], page 2,[copy-pass], page 6]
Reset the access times of files after reading them, so that it does not look like they have just been read.

`-A`
`--append` [[copy-out], page 2]
Append to an existing archive. Only works in copy-out mode. The archive must be a disk file specified with the `'-0'` or `'-F'` (`--file`) option.

`-b`
`--swap` [[copy-in], page 4]
Swap both halfwords of words and bytes of halfwords in the data. Equivalent to `'-sS'`. This option may be used in copy-in mode. Use this option to convert 32-bit integers between big-endian and little-endian machines.

`-B` [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Set the I/O block size to 5120 bytes. Initially the block size is 512 bytes.

`--block-size=block-size` [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Set the I/O block size to `block-size * 512` bytes.

-c [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Use the old portable (ASCII) archive format.

-C *io-size*
--io-size=*io-size*
 [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Set the I/O block size to *io-size* bytes.

-d
--make-directories
 [[copy-in], page 4,[copy-pass], page 6]
Create leading directories where needed.

-D *dir*
--directory=*dir*
 [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
Change to the directory *dir* before starting the operation. This can be used, for example, to extract an archive contents in a different directory:

```
$ cpio -i -D /usr/local < archive
```

or to copy-pass files from one directory to another:

```
$ cpio -D /usr/bin -p /usr/local/bin < filelist
```

The ‘-D’ option does not affect file names supplied as arguments to another command line options, such as ‘-F’ or ‘-E’. For example, the following invocation:

```
cpio -D /tmp/foo -d -i -F arc
```

instructs `cpio` to open the archive file ‘`arc`’ in the current working directory, then change to the directory ‘`/tmp/foo`’ and extract files to that directory. If ‘`/tmp/foo`’ does not exist, it will be created first (the ‘-d’ option) and then changed to.

-E *file*
--pattern-file=*file*
 [[copy-in], page 4]
Read additional patterns specifying filenames to extract or list from *file*. The lines of *file* are treated as if they had been non-option arguments to `cpio`. This option is used in copy-in mode,

-f
--nonmatching
 [[copy-in], page 4]
Only copy files that do not match any of the given patterns.

-F *archive*
--file=*archive*
 [[copy-in], page 4,[copy-out], page 2]
Archive filename to use instead of standard input or output. To use a tape drive on another machine as the archive, use a filename that starts with ‘`hostname:`’, where *hostname* is the name or IP address of the machine. The hostname can be preceded by a username and an ‘@’ to access the remote tape drive as that user,

if you have permission to do so (typically an entry in that user's '~/.rhosts' file).

--force-local

[[[copy-in](#)], [page 4](#),[copy-out](#)], [page 2](#)]

With '-F', '-I', or '-O', take the archive file name to be a local file even if it contains a colon, which would ordinarily indicate a remote host name.

-H *format*

--format=*format*

[[[copy-in](#)], [page 4](#),[copy-out](#)], [page 2](#),[copy-pass](#)], [page 6](#)]

Use archive format *format*. The valid formats are listed below with file size limits for individual files in parentheses; the same names are also recognized in all-caps. The default in copy-in mode is to automatically detect the archive format, and in copy-out mode is 'bin'.

'bin' The obsolete binary format. (2147483647 bytes)

'odc' The old (POSIX.1) portable format. (8589934591 bytes)

'newc' The new (SVR4) portable format, which supports file systems having more than 65536 i-nodes. (4294967295 bytes)

'crc' The new (SVR4) portable format with a checksum added.

'tar' The old tar format. (8589934591 bytes)

'ustar' The POSIX.1 tar format. Also recognizes GNU tar archives, which are similar but not identical. (8589934591 bytes)

'hpbinc' The obsolete binary format used by HPUX's cpio (which stores device files differently).

'hpodc' The portable format used by HPUX's cpio (which stores device files differently).

-i

--extract

Run in copy-in mode. See [Section 3.2 \[Copy-in mode\]](#), [page 4](#).

-I *archive*

[[[copy-in](#)], [page 4](#)]

Archive filename to use instead of standard input. To use a tape drive on another machine as the archive, use a filename that starts with '*hostname*:', where *hostname* is the name or IP address of the remote host. The hostname can be preceded by a username and an '@' to access the remote tape drive as that user, if you have permission to do so (typically an entry in that user's '~/.rhosts' file).

-l

--link

[[[copy-pass](#)], [page 6](#)]

Link files instead of copying them, when possible.

-L
--dereference [[copy-in], page 4],[copy-pass], page 6]
Copy the file that a symbolic link points to, rather than the symbolic link itself.

-m
--preserve-modification-time [[copy-in], page 4],[copy-pass], page 6]
Retain previous file modification times when creating files.

-M message
--message=message [[copy-in], page 4],[copy-out], page 2]
Print *message* when the end of a volume of the backup media (such as a tape or a floppy disk) is reached, to prompt the user to insert a new volume. If *message* contains the string ‘%d’, it is replaced by the current volume number (starting at 1).

-n
--numeric-uid-gid [[copy-in], page 4]
Show numeric UID and GID instead of translating them into names when using the ‘--verbose’ option.

--no-absolute-filenames [[copy-in], page 4],[copy-out], page 2]
Create all files relative to the current directory in copy-in mode, even if they have an absolute file name in the archive.

--no-preserve-owner [[copy-in], page 4],[copy-pass], page 6]
Do not change the ownership of the files; leave them owned by the user extracting them. This is the default for non-root users, so that users on System V don’t inadvertently give away files. This option can be used in copy-in mode and copy-pass mode

-o
--create Run in copy-out mode. See Section 3.1 [Copy-out mode], page 2.

-O archive [[copy-out], page 2]
Archive filename to use instead of standard output. To use a tape drive on another machine as the archive, use a filename that starts with ‘*hostname*:’, where *hostname* is the name or IP address of the machine. The hostname can be preceded by a username and an ‘@’ to access the remote tape drive as that user, if you have permission to do so (typically an entry in that user’s ‘~/rhosts’ file).

--only-verify-crc [[copy-in], page 4]
Verify the CRC’s of each file in the archive, when reading a CRC format archive. Don’t actually extract the files.

-p
--pass-through
 Run in copy-pass mode. See Section 3.3 [Copy-pass mode], page 6.

--quiet [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
 Do not print the number of blocks copied.

-r
--rename [[copy-in], page 4]
 Interactively rename files.

-R owner
--owner owner [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
 In copy-in and copy-pass mode, set the ownership of all files created to the specified *owner* (this operation is allowed only for the super-user). In copy-out mode, store the supplied owner information in the archive.
 The argument can be either the user name or the user name and group name, separated by a dot or a colon, or the group name, preceded by a dot or a colon, as shown in the examples below:

```
cpio --owner smith
cpio --owner smith:
cpio --owner smith:users
cpio --owner :users
```

 The argument parts are first looked up in the system user and group databases, correspondingly. If any of them is not found there, it is treated as numeric UID or GID, provided that it consists of decimal digits only.
 To avoid the lookup and ensure that arguments are treated as numeric values, prefix them with a plus sign, e.g.:

```
cpio --owner +0
cpio --owner +0:
cpio --owner +0:+0
cpio --owner :+0
```

 If the group is omitted but the ':' or '.' separator is given, as in the second example. the given user's login group will be used.

--rsh-command=command [[copy-in], page 4,[copy-out], page 2,[copy-pass], page 6]
 Notifies cpio that it should use *command* to communicate with remote devices.

-s
--swap-bytes [[copy-in], page 4]
 Swap the bytes of each halfword (pair of bytes) in the files. This option can be used in copy-in mode.

-S
--swap-halfwords [[copy-in], page 4]
 Swap the halfwords of each word (4 bytes) in the files. This option may be used in copy-in mode.

- sparse** [\[\[copy-in\], page 4\]](#), [\[copy-pass\], page 6\]](#)
Write files with large blocks of zeros as sparse files. This option is used in copy-in and copy-pass modes.
- t**
- list** [\[\[copy-in\], page 4\]](#)
Print a table of contents of the input. Can be used alone, as a mode designator, in which case `'-i'` is implied.
- to-stdout**
[\[\[copy-in\], page 4\]](#)
Extract files to standard output. This option may be used in copy-in mode.
- u**
- unconditional**
[\[\[copy-in\], page 4\]](#), [\[copy-pass\], page 6\]](#)
Replace all files, without asking whether to replace existing newer files with older files.
- v**
- verbose**
[\[\[copy-in\], page 4\]](#), [\[copy-out\], page 2\]](#), [\[copy-pass\], page 6\]](#)
List the files processed, or with `'-t'`, give an `ls -l` style table of contents listing. In a verbose table of contents of a ustar archive, user and group names in the archive that do not exist on the local system are replaced by the names that correspond locally to the numeric UID and GID stored in the archive.
- V**
- dot** [\[\[copy-in\], page 4\]](#), [\[copy-out\], page 2\]](#), [\[copy-pass\], page 6\]](#)
Print a `'.'` for each file processed.
- version**
Print the `cpio` program version number and exit.
- W**
- warning=flag**
[\[\[copy-in\], page 4\]](#), [\[copy-out\], page 2\]](#), [\[copy-pass\], page 6\]](#)
Control warning display. The argument is one of the following:
- | | |
|--------------------------|--|
| <code>none</code> | Disable all warnings. |
| <code>all</code> | Enable all warnings. |
| <code>truncate</code> | Warn about truncation of file header fields. |
| <code>no-truncate</code> | Disable truncation warnings. |

4 Magnetic Media

Archives are usually written on removable media—tape cartridges, mag tapes, or floppy disks.

The amount of data a tape or disk holds depends not only on its size, but also on how it is formatted. A 2400 foot long reel of mag tape holds 40 megabytes of data when formatted at 1600 bits per inch. The physically smaller EXABYTE tape cartridge holds 2.3 gigabytes.

Magnetic media are re-usable—once the archive on a tape is no longer needed, the archive can be erased and the tape or disk used over. Media quality does deteriorate with use, however. Most tapes or disks should be discarded when they begin to produce data errors.

Magnetic media are written and erased using magnetic fields, and should be protected from such fields to avoid damage to stored data. Sticking a floppy disk to a filing cabinet using a magnet is probably not a good idea.

5 Reporting bugs or suggestions

It is possible you will encounter a bug in `cpio`. If this happens, we would like to hear about it. As the purpose of bug reporting is to improve software, please be sure to include maximum information when reporting a bug. The information needed is:

- Version of the package you are using.
- Compilation options used when configuring the package.
- Conditions under which the bug appears.

Send your report to bug-cpio@gnu.org. This is a public mailing list; all correspondence sent to it is archived and is available at <https://lists.gnu.org/archive/html/bug-cpio>. Your bug report will be visible there, too. The list is not limited to bug reports, in fact it is dedicated to any kind of technical discussions regarding GNU `cpio`. If you wish to subscribe to it, visit <https://lists.gnu.org/mailman/listinfo/bug-cpio>.

Concept Index

A

archive creation 2
archive extraction 4

C

command line options 2
copy files between filesystems 6
copy-in 4
copy-out 2
copy-pass 6
copying directory structures 1
create archive 2
creating a `cpio` archive 1

E

extract files from an archive 4
extracting a `cpio` archive 1

I

invoking `cpio` 2

M

magnetic media 15

P

passing directory structures 1

Table of Contents

1	Introduction	1
2	Tutorial	1
3	Invoking cpio	2
3.1	Copy-out mode	2
3.2	Copy-in mode	4
3.3	Copy-pass mode	6
3.4	Options	9
4	Magnetic Media	15
5	Reporting bugs or suggestions	15
	Concept Index	15