NAME
cpio – copy file archives in and out

SYNOPSIS

cpio −o [−0AaBcJjLVvZz][−C bytes][−F archive][−H format][−M flag][−0 archive]<
name-list[> archive]
cpio −i [−06BbcdfJjmrSstuVvZz][−C bytes][−E file][−F archive][−H format][−I
archive][−M flag][pattern ...]< archive

cpio −p [−0adLlmuVv]destination−directory< name-list

DESCRIPTION

The cpio command copies files to and from a cpio archive.

The options are as follows:

−0 Use the NUL (‘\0’) character as a pathname terminator, instead of newline (‘\n’). This applies only to the pathnames read from standard input in the write and copy modes, and to the path-
names written to standard output in list mode. This option is expected to be used in concert with the
−print0 function in find(1) or the −0 flag in xargs(1).

−o Create an archive. Reads the list of files to store in the archive from standard input, and writes the
archive on standard output.

−A Append to the specified archive.

−a Reset the access times on files that have been copied to the archive.

−B Set block size of output to 5120 bytes.

−C bytes
Set the block size of output to bytes.

−c Use ASCII format for cpio header for portability.

−F archive
Use the specified file as the input for the archive.

−H format
Write the archive in the specified format. Recognised formats are:

ar Unix Archiver.
bcpio Old binary cpio format. Selected by −6.
cpio Old octal character cpio format. Selected by −c.
sv4cpio
SVR4 hex cpio format.
sv4crc
SVR4 hex cpio format with checksums. This is the default format for creating
new archives.
tar Old tar format.
ustar POSIX ustar format.

bin These four formats...
crc ...are supported...
newc ...for backwards...
...compatibility only.

-J  Use the xz utility to compress the archive.

-j  Use the bzip2 utility to compress the archive.

-L  Follow symbolic links.

-M flag
Configure the archive normaliser. *flag* is either a numeric value compatible to `strtonum(3)` which is directly stored in the flags word, or one of the following values, optionally prefixed with “no-” to turn them off:

**inodes**

0x0001: Serialise inodes, zero device info.

(cpio, sv4cpio, sv4crc)

**links**

0x0002: Store content of hard links only once.

(cpio, sv4cpio, sv4crc)

**mtime**

0x0004: Zero out the file modification time.

(ar, cpio, sv4cpio, sv4crc, ustar)

**uid gid**

0x0008: Set owner to 0:0 (root:wheel).

(ar, cpio, sv4cpio, sv4crc, ustar)

**verb**

0x0010: Debug this option.

debug

0x0020: Debug file header storage.

**lncp**

0x0040: Extract hard links by copy if link fails.

**numid**

0x0080: Use only numeric uid and gid values.

(ustar)

**gslash**

0x0100: Append a slash after directory names.

(ustar)

**set**

0x0003: Keep ownership and mtime intact.

**dist**

0x008B: Clean everything except mtime.

**norm**

0x008F: Clean everything.

**root**

0x0089: Clean owner and device information.

When creating an archive and verbosely listing output, these normalisation operations are not reflected in the output, because they are made only after the output has been shown.

This option is only implemented for the ar, cpio, sv4cpio, sv4crc, and ustar file format writing routines.

-O archive

Use the specified file name as the archive to write to.

-V  Print a dot (‘.’) for each file written to the archive.

-v  Be verbose about operations. List filenames as they are written to the archive.

-Z  Use the compress(1) utility to compress the archive.

-z  Use the gzip(1) utility to compress the archive.

-i Restore files from an archive. Reads the archive file from standard input and extracts files matching the *patterns* that were specified on the command line.

-6 Process old-style cpio format archives.
−B Set the block size of the archive being read to 5120 bytes.
−b Do byte and word swapping after reading in data from the archive, for restoring archives
created on systems with a different byte order.
−c bytes
  Read archive written with a block size of bytes.
−c Expect the archive headers to be in ASCII format.
−d Create any intermediate directories as needed during restore.
−E file
  Read list of file name patterns to extract or list from file.
−F archive, −I archive
  Use the specified file as the input for the archive.
−f Restore all files except those matching the patterns given on the command line.
−H format
  Read an archive of the specified format. Recognised formats are:
  ar Unix Archiver.
  bcpio Old binary cpio format.
  cpio Old octal character cpio format.
  sv4cpio
    SVR4 hex cpio format.
  sv4crc
    SVR4 hex cpio format with checksums.
  tar Old tar format.
  ustar POSIX ustar format.

  bin These four formats...
  crc ...are supported...
  newc ...for backwards...
  odc ...compatibility only.
−J Use the xz utility to decompress the archive.
−j Use the bzip2 utility to decompress the archive.
−m Restore modification times on files.
−r Rename restored files interactively.
−S Swap words after reading data from the archive.
−s Swap bytes after reading data from the archive.
−t Only list the contents of the archive, no files or directories will be created.
−u Overwrite files even when the file in the archive is older than the one that will be over-
written.
−V Print a dot (‘.’) for each file read from the archive.
−v Be verbose about operations. List filenames as they are copied in from the archive.
Use the compress(1) utility to decompress the archive.

Use the gzip(1) utility to decompress the archive.

Copy files from one location to another in a single pass. The list of files to copy are read from standard input and written out to a directory relative to the specified directory argument.

Reset the access times on files that have been copied.

Create any intermediate directories as needed to write the files at the new location.

Follow symbolic links.

When possible, link files rather than creating an extra copy.

Restore modification times on files.

Overwrite files even when the original file being copied is older than the one that will be overwritten.

Print a dot (‘.’) for each file copied.

Be verbose about operations. List filenames as they are copied.

**ENVIRONMENT**

TMPDIR Path in which to store temporary files.

**EXIT STATUS**

The cpio utility exits with one of the following values:

0 All files were processed successfully.

1 An error occurred.

**DIAGNOSTICS**

Whenever cpio cannot create a file or a link when extracting an archive or cannot find a file while writing an archive, or cannot preserve the user ID, group ID, file mode, or access and modification times when the -p option is specified, a diagnostic message is written to standard error and a non-zero exit value will be returned, but processing will continue. In the case where cpio cannot create a link to a file, unless -M Incp is given, cpio will not create a second copy of the file.

If the extraction of a file from an archive is prematurely terminated by a signal or error, cpio may have only partially extracted the file the user wanted. Additionally, the file modes of extracted files and directories may have incorrect file bits, and the modification and access times may be wrong.

If the creation of an archive is prematurely terminated by a signal or error, cpio may have only partially created the archive, which may violate the specific archive format specification.

**SEE ALSO**

ar(1), pax(1), tar(1)

**AUTHORS**

Keith Muller at the University of California, San Diego. MirOS extensions by mirabilos <m@mirbsd.org>.

**CAVEATS**

Different file formats have different maximum file sizes. It is recommended that a format such as cpio or ustar be used for larger files.
<table>
<thead>
<tr>
<th>File format</th>
<th>Maximum file size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar</td>
<td>10 Gigabytes - 1 Byte</td>
</tr>
<tr>
<td>bcpio</td>
<td>4 Gibibytes</td>
</tr>
<tr>
<td>sv4cpio</td>
<td>4 Gibibytes</td>
</tr>
<tr>
<td>cpio</td>
<td>8 Gibibytes</td>
</tr>
<tr>
<td>tar</td>
<td>8 Gibibytes</td>
</tr>
<tr>
<td>ustar</td>
<td>8 Gibibytes</td>
</tr>
</tbody>
</table>

The backwards-compatible format options are not available in the pax(1) front-end.

The -M option is a MirBSD extension, available starting with MirOS #8. Archives written using these options are, however, compatible to the standard and should be readable on any other system. The only option whose behaviour is not explicitly allowed by the standard is hard link unification (write file contents only once) selected by -M 0x0002.

The -V option is a GNU extension, available starting with MirOS #11.

The ar file format matches APT repositories and the BSD ar(1) specification, not GNU binutils (which can however read them) or SYSV systems.

**BUGS**

The -s and -S options are currently not implemented.

The pax file format is not yet supported.