GNU CPIO

GNU cpio 2.5 June 2002

by Robert Carleton

This is the first edition of the GNU cpio documentation,
and is consistent with GNU cpio 2.5.

Published by the Free Software Foundation
59 Temple Place - Suite 330,
Boston, MA 02111-1307, 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 compatability 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 because 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 place them in the present directory. 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 to the current directory. 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 3.

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 3.

```
% 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. See Section 3.4 [Options], page 3.

```
cpio {-o|--create} [-0acvABLV] [-C bytes] [-H format]
[-M message] [-O [[user@]host:]archive] [-F [[user@]host:]archive]
[--file=[[user@]host:]archive] [--format=format]
[--message=message][--null] [--reset-access-time] [--verbose]
[--dot] [--append] [--block-size=blocks] [--dereference]
[--io-size=bytes] [--rsh-command=command] [--help] [--version]
< name-list [> archive]
```

3.2 Copy-in mode

In copy-in mode, cpio copies files out of an archive 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. See Section 3.4 [Options], page 3.

```
cpio {-i|--extract} [-bcdfmnrtsuvBSV] [-C bytes] [-E file]
[-H format] [-M message] [-R [user][:.][group]]
[-I [[user@]host:]archive] [-F [[user@]host:]archive]
[--file=[[user@]host:]archive] [--make-directories]
[--nonmatching] [--preserve-modification-time]
[--numeric-uid-gid] [--rename] [--list] [--swap-bytes] [--swap]
[--dot] [--unconditional] [--verbose] [--block-size=blocks]
[--swap-halfwords] [--io-size=bytes] [--pattern-file=file]
[--format=format] [--owner=[user][:.][group]]
[--no-preserve-owner] [--message=message] [--help] [--version]
[-no-absolute-filenames] [--sparse] [-only-verify-crc] [-quiet]
[--rsh-command=command] [pattern...] [< archive]
```

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. See Section 3.4 [Options], page 3.

```
cpio {-p|--pass-through} [-0adlmuvLV] [-R [user][:.][group]]
[--null] [--reset-access-time] [--make-directories] [--link]
[--preserve-modification-time] [--unconditional] [--verbose]
[--dot] [--dereference] [--owner=[user][:.][group]] [--sparse]
[--no-preserve-owner] [--help] [--version] destination-directory
< name-list
```

3.4 Options

`-0, --null`
: 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`
: Reset the access times of files after reading them, so that it does not look like they have just been read.

`-A, --append`
: Append to an existing archive. Only works in copy-out mode. The archive must be a disk file specified with the -O or -F (–file) option.

`-b, --swap`
: 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`
: Set the I/O block size to 5120 bytes. Initially the block size is 512 bytes.

`--block-size=BLOCK-SIZE`
: Set the I/O block size to BLOCK-SIZE * 512 bytes.

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

`-C IO-SIZE, --io-size=IO-SIZE`
: Set the I/O block size to IO-SIZE bytes.

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

`-E FILE, --pattern-file=FILE`
: 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`
: Only copy files that do not match any of the given patterns.

`-F, --file=archive`
: 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:'. 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`
: 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`
: Use archive format FORMAT. The valid formats are listed below; 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.

  '`odc`' The old (POSIX.1) portable format.

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

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

  '`tar`' The old tar format.

  '`ustar`' The POSIX.1 tar format. Also recognizes GNU tar archives, which are similar but not identical.

'hpbin' 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 3.

`-I archive`
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:'. 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).

`-k` Ignored; for compatibility with other versions of cpio.

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

`-L, --dereference`
Copy the file that a symbolic link points to, rather than the symbolic link itself.

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

`-M MESSAGE, --message=MESSAGE`
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`
Show numeric UID and GID instead of translating them into names when using the '--verbose option'.

`--no-absolute-filenames`
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`
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 inadvertantly 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`
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:'. 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`
: 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 3.

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

`-r, --rename`
: Interactively rename files.

`-R [user][:.][group], --owner [user][:.][group]`
: Set the ownership of all files created to the specified user and/or group in copy-out and copy-pass modes. Either the user, the group, or both, must be present. If the group is omitted but the ":" or "." separator is given, use the given user's login group. Only the super-user can change files' ownership.

`--rsh-command=COMMAND`
: Notifies cpio that is should use COMMAND to communicate with remote devices.

`-s, --swap-bytes`
: Swap the bytes of each halfword (pair of bytes) in the files.This option can be used in copy-in mode.

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

`--sparse`
: Write files with large blocks of zeros as sparse files. This option is used in copy-in and copy-pass modes.

`-t, --list`
: Print a table of contents of the input.

`-u, --unconditional`
: Replace all files, without asking whether to replace existing newer files with older files.

`-v, --verbose`
: 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`
: Print a `.` for each file processed.

`--version`
: Print the cpio program version number and exit.

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 formated 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 disgarded 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.

Concept Index

Table of Contents