

# Contents

## Name

pax - portable archive interchange

## Description

The pax utility supports the creation, reading and writing of portable archive files on formatted media such as local hard disks and floppy disks. This version provides support for portable archives recorded in the tar, ustar or cpio formats.

New archives are created using the New Archive option on the File Menu; existing archives are opened using the Open Archive option. When an existing archive is opened, its contents are listed as if pax had been called on a UNIX system with neither "-r" nor "-w" specified. Thereafter, the archive can be (re)written or existing archive files can be selected and extracted.

The choice of which recording format to use is made when an archive is first written. Thereafter the format is fixed until either the archive is deleted or rewritten. Existing archives may have extra files appended to them.

The Read and Write options on the Actions menu are used to extract and create archives. Note that Read and Write will not manage spanned, multi-volume archives directly (see Miscellaneous below).

## See Also

[Writing archives](#)

[Reading archives](#)

[Output display](#)

[Archive formats](#)

[File names](#)

[Miscellaneous](#)

# Writing archives

Archives are written by selecting the Write option on the Actions menu. This menu item is only enabled after either a new archive file has been created or an existing archive has been opened. Selecting Write causes the Write Archive dialog to be displayed, which contains the following fields:

## **Source Directory:**

Selects the directory containing the file or files to be archived. This can be specified either by typing the directory name directly, or by using the Browse button to scan the directory hierarchy of your local disks.

## **Files:**

Identifies the file or files to be archived as one or more space separated filename patterns, possibly containing DOS wildcard characters. For example, to archive all C++ source and header files, this field could contain "\*.cpp \*.h". Note that only one filename pattern can be specified in unregistered versions of the software.

## **Descend Subdirectories:**

If checked, pax will scan all subdirectories under the specified source directory for matching files. This also causes directory nodes themselves to be archived. So, for example, to archive an entire directory structure, select the appropriate source directory, set Files to "\*\*.\*", and check Descend Subdirectories.

## **Format**

This field allows the user to select the archive format; that is, one of Cpio, Ustar or Tar. For a new archive file, Ustar will be set by default; for an existing archive file, the current format will be set. Note that the format of an existing archive can only be changed if the Create option is also specified (see below).

## **Action**

Selecting Create causes the archive to be rewritten from the beginning, thus destroying its existing contents (if any). Selecting Append causes the identified Files to be added to the end of the archive. Judicious use of Append mode allows multiple versions of a file or files to be maintained in a single archive file.

# Reading archives

Archives are read by selecting the Read option on the Actions menu. This menu item is only enabled after either a new archive file has been created or an existing archive has been opened. Selecting Read causes the Read Archive dialog to be displayed, which contains the following fields:

## **Target Directory:**

This field identifies the directory into which the selected files and directories will be read. The target directory can be specified either by typing its name directly, or by using the associated Browse button to scan the directory hierarchy of local disks. Use the Create Dir button to create a new target directory.

## **Extract**

Files and directories can be selected for extraction by using the mouse to select their names on the output display. Pax permits multiple such selections using normal Windows Listbox controls. Alternatively, all the file and directories in an archive can be extracted by selecting All files.

## **Options**

Interactively rename files. If unchecked, files and directory names in the archive will be preserved when extracted. If checked, a dialog will be displayed before each file is extracted, allowing the identified file to be skipped, its name to be retained, or its name to be changed as follows... In the To field of the Rename dialog, type a new name to rename the file, type a dot (.) to preserve its existing name, or leave the To field blank to skip the file.

Do not overwrite existing files. If checked, archive members identifying files that already exist in the target directory will not be extracted.

Ignore older files. If checked, archive members identifying older versions of files that already exist in the target directory will not be extracted. Newer versions will be extracted as normal and will replace older files in the target directory.

Restore access times. If unchecked, the access times of files extracted from an archive will be set to the time at which they are extracted. If checked, access times will be restored from the archive.

# Output display

Archive members are listed in the output window, one per line, in the following format:

```
000666 1 0 0 204 06-08-1995 stdafx.cpp
000666 1 0 0 283 06-08-1995 stdafx.h
000777 1 0 0 0 04-02-1996 hlp/
000666 1 0 0 0 26-02-1996 hlp/setup.hh
000666 1 0 0 834 26-02-1996 hlp/setup.hpj
000666 1 0 0 25673 26-02-1996 hlp/setup.vh
```

The first column in this display contains the "mode" field, which varies according to the recording format of the archive. Note that this is the UNIX mode of the file, artificially generated under Windows to provide the best approximation. The octal values recorded in this field have the following meanings:

```
100000 Regular file (cpio only)
060000 Block special file (cpio only)
040000 Directory (cpio only)
020000 Character special file (cpio only)
010000 FIFO special file (cpio only)

004000 Set UID on execution
002000 Set GID on execution
001000 Sticky bit

000700 Read, write and execute owner permissions
000070 Read, write and execute group permissions
000007 Read, write and execute other permissions
```

The second column contains the link count, which is meaningless under Windows and will almost always be set to 1.

The third and fourth column contain the User ID and Group ID respectively, which again are not meaningful under Windows. An archive created under Windows will always contain zeros in these fields.

The fifth column contains the size of the associated archive member in bytes. The sixth column contains the access time of the file (mtime) in the form dd-mm-yyyy, and the final column contains the pathname of the file, again in UNIX format.

Note that file names in DOS/Windows systems are case insensitive, whereas UNIX file names are case sensitive. This may cause problems where files are archived under UNIX, transferred to Windows, extracted locally, re-archived and then returned to their original UNIX host. This process will return all names in lower case (i.e., possibly dropping upper case characters from the original names).

# Archive formats

This version of pax supports the following portable archive formats:

## **Cpio**

Extended cpio interchange format recommended by X/Open for interchange of information between systems. This format imposes no restrictions on the number of characters in a path name.

## **Ustar**

Extended tar interchange format recommended by POSIX (and hence ISO/IEC Std 9945) for interchange of information between systems. This format imposes a limit of 256 characters in a path name.

## **Tar**

An old but still common interchange format dating from pre-POSIX compliant UNIX systems. This format imposes a limit of 100 characters in a path name, and for this reason is not recommended for general use on Windows95 and Windows NT systems.

## **Miscellaneous Info**

The older cpio binary format is not supported, as this format is not generally portable between systems having different basic data types (e.g., big endian vs little endian).

Many file types supported under UNIX are neither meaningful nor supportable under Windows (e.g., hard links, symbolic links, character special files, block special files, FIFO special files, etc.). These are ignored when encountered by this version of the pax utility.

Note that the theoretical file size limit in all three of the above archive formats is 8 gigabytes. If anyone achieves this will they please let me know.

# File Names

On modern UNIX systems, file names and path names can be virtually any length, up to implementation limits defined by the constants `PATH_MAX` and `NAME_MAX`. All POSIX conforming systems must support values of at least 255 and 14 respectively for these constants, which are the minimum values (in bytes) that shall apply across all supported file-systems. The actual limit per file-system type is configurable and may exceed the above values.

Clearly, this can mean that these names are not recreatable from tar or cpio archives processed on Windows systems. Both Windows 3.x and Windows95 systems have absolute path limits of 260 characters. Windows 3.x, or more precisely the underlying DOS system, further limits elements within a path name to the classic 8.3 format. Windows95 does not have this latter limitation.

This problem is alleviated slightly by the fact that the tar format limits path names to 100 characters, and the ustar format limits them to 256 characters. Hence all possible path name settings should be supportable under Windows95. Note, however, that the cpio format has no such limitations and path names can be any size up to the limit supported by individual implementations.

For Windows 3.x systems, it is necessary to process path names such that they are made into valid DOS path names, which the 16-bit version of WinXs pax does using the following simple algorithm:

- First, a complete path name is broken down into its constituent elements.
- Each element name is reduced to 8.3 format by taking the first 8 non-space characters before the first period, followed by a period, followed by the first 3 characters after the last period in the input name. The last two elements (i.e., the period and extension) will be omitted if no period appears in the input name.
- Finally, the mangled 8.3 element names are reformed into a complete path name, which must now form a valid DOS path name.

If the total length of the resulting path name is still larger than 260 characters, the read request will fail.

While this process ensures that the path names of files extracted from portable archives are valid in a DOS environment, it does raise the possibility of generating like names for different files. This problem can be overcome by interactively renaming files during extraction. Note also that because all option checking is performed after name mangling, the "Do not overwrite existing files" option can be used to prevent successive files with the same generated name from overwriting one another.

# Miscellaneous

## Spanned Archives

Spanned (multi-volume) archives are not supported directly by this version of pax. Neither X/Open nor POSIX specify standards for spanned archives and any mechanism invented locally could make it difficult or even impossible to read WinXs archives using other implementations of pax, cpio or tar.

To write a portable archive to multiple floppy disks, first write the archive to a hard disk and then split it using the WinXs split utility, in 2800kb, 1400kb, 720kb or 360kb chunks, depending in the capacity of the floppy disk. This will generate a series of files named <archive>.xaa, <archive>.xab, etc., where <archive> will be the same as the name of the original archive file. Now copy each of these files to a separate floppy disk. To rebuild the archive on another system, copy the above chunks from floppy disk to the local hard disk and concatenate them to a single file using the cat(1) command (or Unsplit if the target system has WinXs installed).

## Unformatted Media

This version of pax supports the creation, reading and writing of extended cpio, ustar and tar archives on formatted media (disks). It may support the reading and writing of archives to certain types of unformatted media such as magnetic tape, but no assurances are given, either implicitly or explicitly, that this product is fit for such a purpose. If you do use such media, note that cpio archives will be written in 5120 byte blocks, ustar and tar archives will be written in 10240 byte blocks, which are the default sizes specified by POSIX.

## Compression

WinPax does not compress portable archives automatically, again because this behaviour is not specified by X/Open or POSIX. Any portable archive can be compressed using the WinXs compress utility before transfer to another system, although note that the output from compress is itself not guaranteed to be portable to all platforms (though it generally is). Similarly, gzip compression can be used to compress an archive and is generally portable to most UNIX systems. Note that Windows-style ZIP compression is infrequently supported on UNIX platforms and generally should not be used to compress portable archives.





