NAME

pax - read and write file archives and copy directory hierarchies

SYNOPSIS

pax [**-cdnvzO**] [**-f** *archive*] [**-s** *replstr*] ... [**-U** *user*] ... [**-G** *group*] ... [**-T** [*from_date*] [,*to_date*]] ... [*pattern ...*]

pax -r [**-cdiknuvzDOYZ**] [**-f** *archive*] [**-o** *options*] ... [**-p** *string*] ... [**-s** *replstr*] ... [**-E** *limit*] [**-U** *user*] ... [**-G** *group*] ... [**-T** [*from_date*] [,*to_date*]] ... [*pattern ...*]

pax -w [**-dituvzHLOPX**] [**-b** *blocksize*] [[**-a**] [**-f** *archive*]] [**-x** *format*] [**-s** *replstr*] ... [**-o** *options*] ... [**-U** *user*] ... [**-G** *group*] ... [**-B** *bytes*] [**-T** [*from_date*] [,*to_date*] [*/[c][m]*]] ... [*file ...*]

pax -r -w [**-diklntuvDHLOPXYZ**] [**-p** *string*] ... [**-s** *replstr*] ... [**-U** *user*] ... [**-G** *group*] ... [**-T** [*from_date*] [,*to_date*] [*/[c][m]*]] ... [*file ...*] *directory*

DESCRIPTION

The **pax** utility will read, write, and list the members of an archive file, and will copy directory hierarchies. These operations are independent of the specific archive format, and support a wide variety of different archive formats. A list of supported archive formats can be found under the description of the **-x** option.

The presence of the **-r** and the **-w** options specifies which of the following functional modes **pax** will operate under: *list*, *read*, *write*, and *copy*.

<none>
> *List*. Write to standard output a table of contents of the members of the archive file read from standard input, whose pathnames match the specified *patterns*. The table of contents contains one filename per line and is written using single line buffering.

-r
> *Read*. Extract the members of the archive file read from the standard input, with pathnames matching the specified *patterns*. The archive format and blocking is automatically determined on input. When an extracted file is a directory, the entire file hierarchy rooted at that directory is extracted. All extracted files are created relative to the current file hierarchy. The setting of ownership, access and modification times, and file mode of the extracted files are discussed in more detail under the **-p** option.

-w
> *Write*. Write an archive containing the *file* operands to standard output using the specified archive format. When no *file* operands are specified, a list of files to copy with one per line is read from standard input. When a *file* operand is also a directory, the entire file hierarchy rooted at that directory will be included.

-r -w
> *Copy*. Copy the *file* operands to the destination *directory*. When no *file* operands are specified,

a list of files to copy with one per line is read from the standard input. When a *file* operand is also a directory the entire file hierarchy rooted at that directory will be included. The effect of the *copy* is as if the copied files were written to an archive file and then subsequently extracted, except that there may be hard links between the original and the copied files (see the **-l** option below).

Warning: The destination *directory* must not be one of the *file* operands or a member of a file hierarchy rooted at one of the *file* operands. The result of a *copy* under these conditions is unpredictable.

While processing a damaged archive during a *read* or *list* operation, **pax** will attempt to recover from media defects and will search through the archive to locate and process the largest number of archive members possible (see the **-E** option for more details on error handling).

OPERANDS

The *directory* operand specifies a destination directory pathname. If the *directory* operand does not exist, or it is not writable by the user, or it is not of type directory, **pax** will exit with a non-zero exit status.

The *pattern* operand is used to select one or more pathnames of archive members. Archive members are selected using the pattern matching notation described by fnmatch(3). When the *pattern* operand is not supplied, all members of the archive will be selected. When a *pattern* matches a directory, the entire file hierarchy rooted at that directory will be selected. When a *pattern* operand does not select at least one archive member, **pax** will write these *pattern* operands in a diagnostic message to standard error and then exit with a non-zero exit status.

The *file* operand specifies the pathname of a file to be copied or archived. When a *file* operand does not select at least one archive member, **pax** will write these *file* operand pathnames in a diagnostic message to standard error and then exit with a non-zero exit status.

OPTIONS

The following options are supported:

-r   Read an archive file from standard input and extract the specified *files*. If any intermediate directories are needed in order to extract an archive member, these directories will be created as if mkdir(2) was called with the bitwise inclusive OR of S_IRWXU, S_IRWXG, and S_IRWXO as the mode argument. When the selected archive format supports the specification of linked files and these files cannot be linked while the archive is being extracted, **pax** will write a diagnostic message to standard error and exit with a non-zero exit status at the completion of operation.

-w   Write files to the standard output in the specified archive format. When no *file* operands are specified, standard input is read for a list of pathnames with one per line without any leading or trailing <blanks>.

-a   Append *files* to the end of an archive that was previously written. If an archive format is not specified with a **-x** option, the format currently being used in the archive will be selected. Any attempt to append to an archive in a format different from the format already used in the archive will cause **pax** to exit immediately with a non-zero exit status. The blocking size used in the archive volume where writing starts will continue to be used for the remainder of that archive volume.

Warning: Many storage devices are not able to support the operations necessary to perform an append operation. Any attempt to append to an archive stored on such a device may damage the archive or have other unpredictable results. Tape drives in particular are more likely to not support an append operation. An archive stored in a regular file system file or on a disk device will usually support an append operation.

-b *blocksize*
        When *writing* an archive, block the output at a positive decimal integer number of bytes per write to the archive file. The *blocksize* must be a multiple of 512 bytes with a maximum of 64512 bytes. A *blocksize* larger than 32256 bytes violates the POSIX standard and will not be portable to all systems. A *blocksize* can end with k or b to specify multiplication by 1024 (1K) or 512, respectively. A pair of *blocksizes* can be separated by x to indicate a product. A specific archive device may impose additional restrictions on the size of blocking it will support. When blocking is not specified, the default *blocksize* is dependent on the specific archive format being used (see the **-x** option).

-c   Match all file or archive members *except* those specified by the *pattern* and *file* operands.

-d   Cause files of type directory being copied or archived, or archive members of type directory being extracted, to match only the directory file or archive member and not the file hierarchy rooted at the directory.

-f *archive*
        Specify *archive* as the pathname of the input or output archive, overriding the default standard input (for *list* and *read*) or standard output (for *write*). A single archive may span multiple files and different archive devices. When required, **pax** will prompt for the pathname of the file or device of the next volume in the archive.

-i   Interactively rename files or archive members. For each archive member matching a *pattern*

operand or each file matching a *file* operand, **pax** will prompt to */dev/tty* giving the name of the file, its file mode and its modification time. The **pax** utility will then read a line from */dev/tty*. If this line is blank, the file or archive member is skipped. If this line consists of a single period, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line. The **pax** utility will immediately exit with a non-zero exit status if <EOF> is encountered when reading a response or if */dev/tty* cannot be opened for reading and writing.

-k   Do not overwrite existing files.

-l   Link files. (The letter ell). In the *copy* mode (**-r -w**), hard links are made between the source and destination file hierarchies whenever possible.

-n   Select the first archive member that matches each *pattern* operand. No more than one archive member is matched for each *pattern*. When members of type directory are matched, the file hierarchy rooted at that directory is also matched (unless **-d** is also specified).

-o *options*
       Information to modify the algorithm for extracting or writing archive files which is specific to the archive format specified by **-x**. In general, *options* take the form: **name=value**

-p *string*
       Specify one or more file characteristic options (privileges). The *string* option-argument is a string specifying file characteristics to be retained or discarded on extraction. The string consists of the specification characters **a**, **e**, **m**, **o**, and **p**. Multiple characteristics can be concatenated within the same string and multiple **-p** options can be specified. The meaning of the specification characters are as follows:

       a   Do not preserve file access times. By default, file access times are preserved whenever possible.

       e   'Preserve everything', the user ID, group ID, file mode bits, file access time, and file modification time. This is intended to be used by *root*, someone with all the appropriate privileges, in order to preserve all aspects of the files as they are recorded in the archive. The **e** flag is the sum of the **o** and **p** flags.

       m   Do not preserve file modification times. By default, file modification times are preserved whenever possible.

       o   Preserve the user ID and group ID.

p　'Preserve' the file mode bits. This intended to be used by a *user* with regular privileges who wants to preserve all aspects of the file other than the ownership. The file times are preserved by default, but two other flags are offered to disable this and use the time of extraction instead.

In the preceding list, 'preserve' indicates that an attribute stored in the archive is given to the extracted file, subject to the permissions of the invoking process. Otherwise the attribute of the extracted file is determined as part of the normal file creation action. If neither the **e** nor the **o** specification character is specified, or the user ID and group ID are not preserved for any reason, **pax** will not set the S_ISUID (*setuid*) and S_ISGID (*setgid*) bits of the file mode. If the preservation of any of these items fails for any reason, **pax** will write a diagnostic message to standard error. Failure to preserve these items will affect the final exit status, but will not cause the extracted file to be deleted. If the file characteristic letters in any of the string option-arguments are duplicated or conflict with each other, the one(s) given last will take precedence. For example, if

　　　-p *eme*

is specified, file modification times are still preserved.

File flags set by chflags(1) are not understood by **pax**, however tar(1) and dump(8) will preserve these.

-s *replstr*

Modify the file or archive member names specified by the *pattern* or *file* operands according to the substitution expression *replstr*, using the syntax of the ed(1) utility regular expressions. The format of these regular expressions are:

　　　/old/new/[gp]

As in ed(1), **old** is a basic regular expression and **new** can contain an ampersand (&), \n (where n is a digit) back-references, or subexpression matching. The **old** string may also contain <newline> characters. Any non-null character can be used as a delimiter (/ is shown here). Multiple **-s** expressions can be specified. The expressions are applied in the order they are specified on the command line, terminating with the first successful substitution. The optional trailing **g** continues to apply the substitution expression to the pathname substring which starts with the first character following the end of the last successful substitution. The first unsuccessful substitution stops the operation of the **g** option. The optional trailing **p** will cause the final result of a successful substitution to be written to standard error in the following format:

　　　<original pathname> >> <new pathname>

File or archive member names that substitute to the empty string are not selected and will be skipped.

-t　Reset the access times of any file or directory read or accessed by **pax** to be the same as they were before being read or accessed by **pax**.

-u   Ignore files that are older (having a less recent file modification time) than a pre-existing file or archive member with the same name. During *read*, an archive member with the same name as a file in the file system will be extracted if the archive member is newer than the file. During *write*, a file system member with the same name as an archive member will be written to the archive if it is newer than the archive member. During *copy*, the file in the destination hierarchy is replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer.

-v   During a *list* operation, produce a verbose table of contents using the format of the ls(1) utility with the **-l** option. For pathnames representing a hard link to a previous member of the archive, the output has the format:

      <ls -l listing> == <link name>

For pathnames representing a symbolic link, the output has the format:

      <ls -l listing> => <link name>

Where <ls -l listing> is the output format specified by the ls(1) utility when used with the **-l** option. Otherwise for all the other operational modes (*read*, *write*, and *copy*), pathnames are written and flushed to standard error without a trailing <newline> as soon as processing begins on that file or archive member. The trailing <newline>, is not buffered, and is written only after the file has been read or written.

-x *format*

    Specify the output archive format, with the default format being *ustar*. The **pax** utility currently supports the following formats:

cpio   The extended cpio interchange format specified in the IEEE Std 1003.2 ("POSIX.2") standard. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by **pax** and is repaired.

bcpio   The old binary cpio format. The default blocksize for this format is 5120 bytes. This format is not very portable and should not be used when other formats are available. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by **pax** and is repaired.

sv4cpio The System V release 4 cpio. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file hard links by this format) which may be truncated by this format is detected by **pax** and is repaired.

sv4crc   The System V release 4 cpio with file crc checksums. The default blocksize for this format is 5120 bytes. Inode and device information about a file (used for detecting file

hard links by this format) which may be truncated by this format is detected by **pax** and is repaired.

tar   The old BSD tar format as found in 4.3BSD. The default blocksize for this format is 10240 bytes. Pathnames stored by this format must be 100 characters or less in length. Only *regular* files, *hard links*, *soft links*, and *directories* will be archived (other file system types are not supported). For backwards compatibility with even older tar formats, a **-o** option can be used when writing an archive to omit the storage of directories. This option takes the form:
          -o write_opt=nodir

ustar   The extended tar interchange format specified in the IEEE Std 1003.2 ("POSIX.2") standard. The default blocksize for this format is 10240 bytes. Pathnames stored by this format must be 255 characters or less in length. The directory part may be at most 155 characters and each path component must be less than 100 characters.

The **pax** utility will detect and report any file that it is unable to store or extract as the result of any specific archive format restrictions. The individual archive formats may impose additional restrictions on use. Typical archive format restrictions include (but are not limited to): file pathname length, file size, link pathname length and the type of the file.

-z   Use gzip(1) to compress (decompress) the archive while writing (reading). Incompatible with **-a**.

-B *bytes*
          Limit the number of bytes written to a single archive volume to *bytes*. The *bytes* limit can end with m, k, or b to specify multiplication by 1048576 (1M), 1024 (1K) or 512, respectively. A pair of *bytes* limits can be separated by x to indicate a product.

          Note that the specified size is for the uncompressed pax image itself. If the **-z** option is also used, the resulting file may contain fewer *bytes*, according to the compressibility of the archive contents. See zip(1) (*ports/archivers/zip*) if compressed volumes of predictable size are required.

          Warning: Only use this option when writing an archive to a device which supports an end of file read condition based on last (or largest) write offset (such as a regular file or a tape drive). The use of this option with a floppy or hard disk is not recommended.

-D   This option is the same as the **-u** option, except that the file inode change time is checked instead of the file modification time. The file inode change time can be used to select files whose inode information (e.g., uid, gid, etc.) is newer than a copy of the file in the destination *directory*.

-E *limit*

> Limit the number of consecutive read faults while trying to read a flawed archives to *limit*. With a positive *limit*, **pax** will attempt to recover from an archive read error and will continue processing starting with the next file stored in the archive. A *limit* of 0 will cause **pax** to stop operation after the first read error is detected on an archive volume. A *limit* of NONE will cause **pax** to attempt to recover from read errors forever. The default *limit* is a small positive number of retries.
>
> *Warning*: Using this option with NONE should be used with extreme caution as **pax** may get stuck in an infinite loop on a very badly flawed archive.

-G *group*

> Select a file based on its *group* name, or when starting with a **#**, a numeric gid. A '\' can be used to escape the **#**. Multiple **-G** options may be supplied and checking stops with the first match.

-H   Follow only command line symbolic links while performing a physical file system traversal.

-L   Follow all symbolic links to perform a logical file system traversal.

-O   Force the archive to be one volume. If a volume ends prematurely, **pax** will not prompt for a new volume. This option can be useful for automated tasks where error recovery cannot be performed by a human.

-P   Do not follow symbolic links, perform a physical file system traversal. This is the default mode.

-T *[from_date][,to_date][/[c][m]]*

> Allow files to be selected based on a file modification or inode change time falling within a specified time range of *from_date* to *to_date* (the dates are inclusive). If only a *from_date* is supplied, all files with a modification or inode change time equal to or younger are selected. If only a *to_date* is supplied, all files with a modification or inode change time equal to or older will be selected. When the *from_date* is equal to the *to_date*, only files with a modification or inode change time of exactly that time will be selected.
>
> When **pax** is in the *write* or *copy* mode, the optional trailing field *[c][m]* can be used to determine which file time (inode change, file modification or both) are used in the comparison. If neither is specified, the default is to use file modification time only. The *m* specifies the comparison of file modification time (the time when the file was last written). The *c* specifies the comparison of inode change time (the time when the file inode was last changed; e.g., a change of owner, group, mode, etc). When *c* and *m* are both specified, then the modification and inode change times are both compared. The inode change time comparison is useful in selecting files whose attributes were recently changed or selecting files which were recently created and had their modification

time reset to an older time (as what happens when a file is extracted from an archive and the modification time is preserved). Time comparisons using both file times is useful when **pax** is used to create a time based incremental archive (only files that were changed during a specified time range will be archived).

A time range is made up of six different fields and each field must contain two digits. The format is:

   [[[[[cc]yy]mm]dd]HH]MM[.SS]

Where *cc* is the first two digits of the year (the century), *yy* is the last two digits of the year, the first *mm* is the month (from 01 to 12), *dd* is the day of the month (from 01 to 31), *HH* is the hour of the day (from 00 to 23), *MM* is the minute (from 00 to 59), and *SS* is the seconds (from 00 to 59). The minute field *MM* is required, while the other fields are optional and must be added in the following order: *HH*, *dd*, *mm*, *yy*, *cc*. The **ss** field may be added independently of the other fields. Time ranges are relative to the current time, so
   -T *1234/cm*
would select all files with a modification or inode change time of 12:34 PM today or later. Multiple **-T** time range can be supplied and checking stops with the first match.

-U *user*
Select a file based on its *user* name, or when starting with a **#**, a numeric uid. A '\' can be used to escape the **#**. Multiple **-U** options may be supplied and checking stops with the first match.

-X   When traversing the file hierarchy specified by a pathname, do not descend into directories that have a different device ID. See the st_dev field as described in stat(2) for more information about device ID's.

-Y   This option is the same as the **-D** option, except that the inode change time is checked using the pathname created after all the file name modifications have completed.

-Z   This option is the same as the **-u** option, except that the modification time is checked using the pathname created after all the file name modifications have completed.

The options that operate on the names of files or archive members (**-c**, **-i**, **-n**, **-s**, **-u**, **-v**, **-D**, **-G**, **-T**, **-U**, **-Y**, and **-Z**) interact as follows.

When extracting files during a *read* operation, archive members are 'selected', based only on the user specified pattern operands as modified by the **-c**, **-n**, **-u**, **-D**, **-G**, **-T**, **-U** options. Then any **-s** and **-i** options will modify in that order, the names of these selected files. Then the **-Y** and **-Z** options will be

applied based on the final pathname. Finally the **-v** option will write the names resulting from these modifications.

When archiving files during a *write* operation, or copying files during a *copy* operation, archive members are 'selected', based only on the user specified pathnames as modified by the **-n**, **-u**, **-D**, **-G**, **-T**, and **-U** options (the **-D** option only applies during a copy operation). Then any **-s** and **-i** options will modify in that order, the names of these selected files. Then during a *copy* operation the **-Y** and the **-Z** options will be applied based on the final pathname. Finally the **-v** option will write the names resulting from these modifications.

When one or both of the **-u** or **-D** options are specified along with the **-n** option, a file is not considered selected unless it is newer than the file to which it is compared.

EXIT STATUS

The **pax** utility will exit with one of the following values:

0   All files were processed successfully.

1   An error occurred.

EXAMPLES

The command:
      pax -w -f /dev/sa0 .
copies the contents of the current directory to the device */dev/sa0*.

The command:
      pax -v -f filename
gives the verbose table of contents for an archive stored in *filename*.

The following commands:
      mkdir /tmp/to
      cd /tmp/from
      pax -rw . /tmp/to
will copy the entire */tmp/from* directory hierarchy to */tmp/to*.

The command:
      pax -r -s ',^//*usr//*,,' -f a.pax
reads the archive *a.pax*, with all files rooted in ''/usr'' into the archive extracted relative to the current directory.

The command:

    pax -rw -i . dest_dir

can be used to interactively select the files to copy from the current directory to *dest_dir*.

The command:

    pax -r -pe -U root -G bin -f a.pax

will extract all files from the archive *a.pax* which are owned by *root* with group *bin* and will preserve all file permissions.

The command:

    pax -r -w -v -Y -Z home /backup

will update (and list) only those files in the destination directory */backup* which are older (less recent inode change or file modification times) than files with the same name found in the source file tree *home*.

DIAGNOSTICS

Whenever **pax** cannot create a file or a link when reading an archive or cannot find a file when writing an archive, or cannot preserve the user ID, group ID, or file mode when the **-p** option is specified, a diagnostic message is written to standard error and a non-zero exit status will be returned, but processing will continue. In the case where pax cannot create a link to a file, **pax** 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, **pax** may have only partially extracted a 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, **pax** may have only partially created the archive which may violate the specific archive format specification.

If while doing a *copy*, **pax** detects a file is about to overwrite itself, the file is not copied, a diagnostic message is written to standard error and when **pax** completes it will exit with a non-zero exit status.

SEE ALSO

cpio(1), tar(1)

STANDARDS

The **pax** utility is a superset of the IEEE Std 1003.2 ("POSIX.2") standard. The options **-z**, **-B**, **-D**, **-E**, **-G**, **-H**, **-L**, **-O**, **-P**, **-T**, **-U**, **-Y**, **-Z**, the archive formats *bcpio*, *sv4cpio*, *sv4crc*, *tar*, and the flawed archive handling during *list* and *read* operations are extensions to the POSIX standard.

HISTORY

The **pax** utility appeared in 4.4BSD.

AUTHORS

Keith Muller at the University of California, San Diego

BUGS

The **pax** utility does not recognize multibyte characters.

File flags set by chflags(1) are not preserved by **pax**. The BUGS section of chflags(1) has a list of utilities that are unaware of flags.