NAME

tar – create tape archives, and add or extract files

SYNOPSIS

/usr/sbin/tar c [**bBefFhilvwX** [**0-7**]] [*device*] [*block*] [*exclude-file* . . .]
[–**I** *include-file*] *file* . . . [–**C** *directory file*]

/usr/sbin/tar r [**bBefFhilvw** [**0-7**]] [*device*] [*block*] [–**I** *include-file*] *file* . . .
[–**C** *directory file*]

/usr/sbin/tar t [**BefFhilvX** [**0-7**]] [*device*] [*exclude-file* . . .] [–**I** *include-file*] [*file* . . .]

/usr/sbin/tar u [**bBefFhilvw** [**0-7**]] [*device*] [*block*] *file* . . .

/usr/sbin/tar x [**BefFhilmopvwX** [**0-7**]] [*device*] [*exclude-file* . . .] [*file* . . .]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **tar** command archives and extracts files to and from a single file called a *tarfile*. A tarfile is usually a magnetic tape, but it can be any file. **tar**'s actions are controlled by the *key* argument. The *key* is a string of characters containing exactly one function letter (**c**, **r**, **t** , **u**, or **x**) and one or more function modifiers, depending on the function letter used. Other arguments to the command are *file*s (or directory names) specifying which files are to be archived or extracted. In all cases, appearance of a directory name refers to the files and (recursively) subdirectories of that directory.

OPERANDS

The following operands are supported:

Function Letters

The function portion of the key is specified by one of the following letters:

c Create. Writing begins at the beginning of the tarfile, instead of at the end. This key implies the **r** key.

r Replace. The named *file*s are written on the end of the tape. The **c** and **u** functions imply this function. See **NOTES** for more information.

t Table of Contents. The names of the specified files are listed each time they occur on the *tarfile*. If no *file*s arguments are given, all the names on the *tarfile* are listed. With the **v** function modifier, additional information for the specified files is displayed. The listing is similar to the format produced by the **ls -l** command.

u Update. The named *file*s are added to the tarfile if they are not already there, or have been modified since last written on that tarfile. This key implies the **r** key. See **NOTES** for more information.

x Extract, or restore. The named *file*s are extracted from the tarfile and written to the current directory. If a named file matches a directory whose contents had been written onto the tarfile, this directory is (recursively) extracted. Use the file or directory's relative path when appropriate, or **tar** will not find a match. The owner, modification time, and mode are restored (if possible); otherwise, to restore owner, you must be the superuser. If no *file*s argument is given, the entire content of the tarfile is extracted. Note: If several files with the same name are on the tarfile, the last one overwrites all earlier ones. See **NOTES** for more information.

Function Modifiers

The characters below may be used in addition to the letter that selects the desired function. Use them in the order shown in the **SYNOPSIS**.

b Blocking Factor. This causes **tar** to use the *block* argument as the blocking factor for tape records. The default is **1**, the maximum is **20**. This function should not be supplied when operating on regular archives or block special devices. It is mandatory however, when reading archives on raw magnetic tape archives (see **f** below). The block size is determined automatically when reading tapes created on block special devices (key letters **x** and **t**). This determination of the blocking factor may be fooled when reading from a pipe or a socket (see the **B** key letter below). The maximum blocking factor is determined only by the amount of memory available to **tar** when it is run. Larger blocking factors result in better throughput, longer blocks on nine-track tapes, and better media utilization.

B Block. Force **tar** to perform multiple reads (if necessary) so as to read exactly enough bytes to fill a block. This option exists so that **tar** can work across the Ethernet, since pipes and sockets return partial blocks even when more data is coming. When reading from standard input, '-', this option is automatically selected to make sure that **tar** can recover from short reads.

e Error. If any unexpected errors occur **tar** will exit immediately with a positive exit status.

f File. This causes **tar** to use the *device* argument as the name of the tarfile. If **f** is given, **/etc/default/tar** is not searched. If **f** is omitted, **tar** will use the device indicated by the **TAPE** environment variable, if set; otherwise, it will use the default values defined in **/etc/default/tar**. If the name of the tarfile is '–', **tar** writes to the standard output or reads from the standard input, whichever is appropriate. Thus, **tar** can be used as the head or tail of a pipeline. **tar** can also be used to move hierarchies with the command:

example% cd fromdir; tar cf – . | (cd todir; tar xfBp –)

F With one **F** argument, **tar** will exclude all directories named SCCS and RCS from the tarfile. With two arguments, **FF**, **tar** will exclude all directories named SCCS and RCS, all files with **.o** as their suffix, and all files named **errs**, **core**, and **a.out**.

h Follow symbolic links as if they were normal files or directories. Normally, **tar** does not follow symbolic links.

i Ignore. With this option **tar** will ignore directory checksum errors.

l Link. This tells **tar** to complain if it cannot resolve all of the links to the files being archived. If **l** is not specified, no error messages are printed.

m Modify. This tells **tar** to not extract the modification times from the tarfile. The modification time of the file will be the time of extraction. This option is only valid with the **x** key.

o Ownership. This causes extracted files to take on the user and group identifier of the user running the program, rather than those on tape. This happens by default for users other than root. If the 'o' option is not set and the user is root, the extracted files will take on the group and user identifiers of the files on tape (see **chown**(1) for more information). The 'o' option is only valid with the **x** key.

p Restore the named files to their original modes, and ACLs if applicable, ignoring the present **umask**(2). SETUID and sticky information are also extracted if your are the super-user. When this option is used with the **c** key letter, ACLs are created in the tarfile along with other information. Note that errors will occur when a tarfile with ACLs is extracted by previous versions of **tar**.

v Verbose. Normally, **tar** does its work silently. This option causes **tar** to type the name of each file it treats, preceded by the function letter. With the **t** function, **v** gives more information about the tape entries than just the name.

w What. This option causes **tar** to print the action to be taken, followed by the name of the file, and then wait for the user's confirmation. If a word beginning with **y** is given, the action is performed. Any other input means no. This is not valid with the **t** key.

X Exclude. Use the *exclude-file* argument as a file containing a list of named files (or directories) to be excluded from the tarfile when using the key letters **c**, **x**, or **t**. Multiple **X** arguments may be used, with one *exclude-file* per argument. See **NOTES** for more information.

[0-7] Select an alternative drive on which the tape is mounted. The default entries are specified in **/etc/default/tar**.

file A path name of a regular file or directory to be archived (when the **c** , **r** or **u** function letters are used), extracted (**x**) or listed (**t**). When *file* is the path name of a directory, the action applies to all of the files and (recursively) subdirectories of that directory. When either or both of the **b** or **f** letters are used in the *key* operand, the initial *file* operands are interpreted as a blocking factor or archive name, as described previously.

If a file name is preceded by –**I** then the file is opened. A list files, one per line, is treated as if each appeared separately on the command line. Be careful of trailing white space in both include and exclude file lists.

In the case where excluded files (see **X** option) also exist, excluded files take precedence over all included files. So, if a file is specified in both the include and exclude files (or on the command line), it will be excluded.

If a file name is preceded by –**C** in a **c** (create) or **r** (replace) operation, **tar** will perform a **chdir** (see **csh**(1)) to that file name. This allows multiple directories not related by a close common parent to be archived using short relative path names.

Note: the –**C** option only applies to *one* following directory name and *one* following file name.

If no digit or 'f' is given, the entry in **/etc/default/tar** with digit "0" will be the default.

EXAMPLES

To archive files from **/usr/include** and from **/etc**, onto default tape drive **0** one might use:

example% tar c –C /usr include –C /etc .

If you get a table of contents from the resulting *tarfile*, you might see something like:

include/
include/a.out.h
and all the other files in **/usr/include** . . .
/chown
and all the other files in **/etc**

To extract all files under **include**:

example% tar xv include
x include/, 0 bytes, 0 tape blocks
and all files under **include**. . .

Here is a simple example using **tar** to create an archive of your home directory on a tape mounted on drive **/dev/rmt/0**:

example% cd
example% tar cvf /dev/rmt/0 .
messages from **tar**

The **c** option means create the archive; the **v** option makes **tar** tell you what it is doing as it works; the **f** option means that you are specifically naming the file onto which the archive should be placed (**/dev/rmt/0** in this example).

Now you can read the table of contents from the archive like this:

```
example% tar tvf /dev/rmt/0
rw-r--r--    1677/40    2123    Nov  7 18:15 1985        ./test.c
. . .
example%
```

The columns have the following meanings:

- column 1 is the access permissions to **./test.c**
- column 2 is the *user-id/group-id* of **./test.c**
- column 3 is the size of **./test.c** in bytes
- column 4 is the modification date of **./test.c**
- column 5 is the name of **./test.c**

You can extract files from the archive like this:

example% tar xvf /dev/rmt/0
messages from **tar**
example%

If there are multiple archive files on a tape, each is separated from the following one by an EOF marker. To have **tar** read the first and second archives from a tape with multiple archives on it, the *non-rewinding* version of the tape device name must be used with the **f** option, as follows:

example% tar xvfp /dev/rmt/0n   *read first archive from tape*

messages from **tar**
example% tar xvfp /dev/rmt/0n *read second archive from tape*
messages from **tar**
example%

(Note that in some earlier releases, the above scenario did not work correctly, and intervention with **mt(1)** between **tar** invocations was necessary.)

Finally, here is an example using **tar** to transfer files across the Ethernet. First, here is how to archive files from the local machine (**example**) to a tape on a remote system (**host**):

example% tar cvfb – 20 *files* **| rsh** *host* **dd of=/dev/rmt/0 obs=20b**
messages from **tar**
example%

In the example above, we are *creating* a *tarfile* with the **c** key letter, asking for *verbose* output from **tar** with the **v** option, specifying the name of the output *tarfile* using the **f** option (the standard output is where the *tarfile* appears, as indicated by the '–' sign), and specifying the blocksize (**20**) with the **b** option. If you want to change the blocksize, you must change the blocksize arguments both on the **tar** command *and* on the **dd** command.

Now, here is how to use **tar** to get files from a tape on the remote system back to the local system:

example% rsh –n host dd if=/dev/rmt/0 bs=20b | tar xvBfb – 20 *files*
messages from **tar**
example%

In the example above, we are *extracting* from the *tarfile* with the **x** key letter, asking for *verbose output from* **tar** with the **v** option, telling **tar** it is reading from a pipe with the **B** option, specifying the name of the input *tarfile* using the **f** option (the standard input is where the *tarfile* appears, as indicated by the '–' sign), and specifying the blocksize (**20**) with the **b** option.

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **tar**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_TIME**, **TZ**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 Successful completion.
>0 An error occurred.

FILES

/dev/rmt/[0-7][b][n]
/dev/rmt/[0-7]l[b][n]
/dev/rmt/[0-7]m[b][n]
/dev/rmt/[0-7]h[b][n]
/dev/rmt/[0-7]u[b][n]
/dev/rmt/[0-7]c[b][n]
/etc/default/tar Settings may look like this:
archive0=/dev/rmt/0
archive1=/dev/rmt/0n

archive2=/dev/rmt/1
archive3=/dev/rmt/1n
archive4=/dev/rmt/0
archive5=/dev/rmt/0n
archive6=/dev/rmt/1
archive7=/dev/rmt/1n

/tmp/tar∗

SEE ALSO

ar(1), **chown**(1), **cpio**(1), **csh**(1), **ls**(1), **mt**(1), **setfacl**(1), **umask**(2), **environ**(5)

DIAGNOSTICS

Complaints about bad key characters and tape read/write errors.
Complaints if enough memory is not available to hold the link tables.

NOTES

There is no way to ask for the *n*-th occurrence of a file.

Tape errors are handled ungracefully.

The **u** option can be slow.

The **b** option should not be used with archives that are going to be updated. The current magnetic tape driver cannot backspace raw magnetic tape. If the archive is on a disk file, the **b** option should not be used at all, because updating an archive stored on disk can destroy it.

The **r** option and the **u** option cannot be used with many tape drives due to limitations in the drive such as the absence of backspace or append capability.

When extracting tapes created with the **r** or **u** option, directory modification times may not be set correctly.

When using **r**,**u**,**x**, or**X**, the named files must match exactly to the corresponding files in the *tarfile*. For example, to extract **./***file*, you must specify **./***file*, and not *file*. The **t** option displays how each file was archived.

The full path name length cannot exceed 255 characters.

The file name (or leaf) length cannot exceed 100 characters.

The prefix of the path name cannot exceed 155 characters.

tar does not copy empty directories or special files such as devices.

Filename substitution wildcards do not work for extracting files from the archive. To get around this, use a command of the form:

tar xvf... /dev/rmt/0 ‘tar tf... /dev/rmt/0 | grep ′*pattern***′‘**

When the Volume Management daemon is running, accesses to floppy devices through the conventional device names (eg, /dev/rdiskette) may not succeed. See **vold**(1M) for further details.