NAME

star − unique standard tape archiver

SYNOPSIS

star *command* [*options*] *file1* . . . *filen*

DESCRIPTION

Star is a tape archiver similar to **tar**(1) but faster than **tar**(1) and with improved functionality.

Star archives and extracts multiple files to and from a single file called a *tarfile.* A tarfile is usually a magnetic tape, but it can be any file. In all cases, appearance of a directory name refers to the files and (recursively) subdirectories of that directory.

Star's actions are controlled by the mandatory command flags from the list below. The way **star** acts may be modified by additional options.

FEATURES

Star uses a fifo to optimize data flow from/to tape. This results in a normally streaming tape during the whole backup. See **−fifo** and **fs=** option to get information on how to find the best fifo size.

Star includes a pattern matcher to control the list of files to be processed. This gives a convenient interface for archiving and restoring complex lists of files. In conjunction with the **−I** flag it is easy to merge a tar archive into an existing file tree. See also **−U** option. In create mode use the **pat=** option to specify select or exclude patterns, in extract or list mode all file type arguments are interpreted as select or exclude patterns. Have a look at the description of the **−C** option to learn how to distribute files to a list of directories.

Star includes a sophisticated diff command. Several diff options allow user tailorable functionality. **Star** won't show you differences you are not interested in. Check the **diffopts=** option for more details.

Star has no limitation on filename length. Pathnames and linknames up to 1024 bytes may be archived. Later versions may be able to deal with longer pathnames.

Star deals with all 3 times, available for files on UNIX systems. **Star** is able to store and restore all 3 times (mtime, atime and even ctime). On Solaris 2.x systems, **star** is able to do backups without changing any of the 3 the times.

If used with the **H=***ustar* option, star is 100% ansi compliant.

Star's new default format *xstar* is as posix compliant as possible. Deviations from the standard that prevent correct extraction with a fully ansi compliant tar implementation occur, when the pathname is longer than 100+130 chars or when archiving sparse files with the −sparse option in effect.

Star makes it easy to repair corrupted filesystems. After a fsck −y has been run on the filesystem, **star** is able to restore only the missing files automatically. Use then *star -diff* to check for differences (see EXAMPLES for more information).

Star automatically recognizes the type of the archive. **Star** therefore is able to handle features and properties of different archive types in their native mode, if it knows about the peculiarities of the archive type. See the **H=**headertype option for more details.

Star automatically recognizes and handles byte swapped archives. There is no option to manually control byte swapping.

COMMAND

−c Create a new *tarfile* and write named files onto it. Writing starts at the beginning of *tarfile.*

−r Replace files in a *tarfile.* The named files are written to the end of *tarfile.* This implies that later, the appropriate files will be found more than once on the *tarfile.* This command is currently not implemented.

−u Update a *tarfile.* The named files are written to the end of *tarfile* if they are not already there. The −r and −u command only work if the tar archives is a regular file or if the tar archive is an unblocked tape that may backspace. This command is currently not implemented.

−x Extract the named files from the *tarfile*. If no filename argument or pattern is specified, the entire content of the *tarfile* is restored. If the −U flag is not used, **star** extracts no file which is older than the corresponding file on disk.

−t Table of contents. List the contents of the *tarfile*. If the −v flag is used, the listing is similar to the format of ls −l output. With this option, the flags **−a**, **−atime** and **−ctime** have a different meaning if the archive is in *star* or *xstar* format. The option **−a** or **−atime** lists the access time instead of the modification time, the option **−ctime** lists the file creation time instead of the modification time.

−n No extraction. Show what **star** would do, in case the **−x** command had been specified.

−diff Diff the contents of the *tarfile* to the filesystem. This may also be used to compare two file trees in the filesystem and gives with a good choice of *diffopts* - in some cases - a more readable output than diff −r. See *diffopts* for more details.

One of the commands above must be specified.

If one or more pattern are specified, they apply to any of the command listed above.

OPTIONS

−help, **−h**
Print a summary of the most important options for **star**(1).

−xhelp Print a summary of the less important options for **star**(1).

blocks=#, **b=**#
Set the blocking factor of the *tarfile* to # times 512 bytes. The default is to use a blocking factor of 20 i.e. 10 kBytes. Increasing the blocksize will speed up the backup. For portability with very old tar implementations (pre BSD 4.2 or pre AT&T SVR4), blocksize should not be more than 10 kBytes. Most systems have a hardware limitation for the blocksize, 32 kBytes and 63 kBytes are common limits on many systems. The upper limit in any case is the size of the buffer RAM in the tape drive. Make a test if you want to make sure that the target system will handle the intended blocksize. If you want to determine the blocking factor of an unknown tar archive on tape, specify a blocking factor that is higher than the supposed blocking factor of the tape. **Star** then will determine the blocking factor by reading the first record of the tape and print a message:

star: Blocksize = # **records.**

Where # is the blocking factor in multiples of 512 bytes. The *blocks=* option is now obsoleted by the *bs=* option described below.

file=tarfilename, **f=**tarfilename
Use *tarfilename* as the name for the tar archive. **Star** normally uses *stdin*/*stdout* for the tar archive because the most common way to use **star** is in conjunction with pipes. If **star** is installed suid root, *tarfilename* may be in remote syntax: user@host:filename as in rcp(1) even if invoked by non root users. See SUID NOTES for more information.

−T If the option **file=** or **f=** is omitted and the −T option is present, **star** will use the device indicated by the **TAPE** environment variable, if set.

−fifo Use a *fifo* to optimize data flow from/to *tarfile*. This option is in effect by default (this may be changed at compile time). The default fifo size is 1MByte. This will even work on a tiny machine like a Sun 3/50. The fifo size may be modified with the *fs=* option. A rule of dumb for the fifo size is to use more than the buffer size of the tape drive and less then half of the real memory of the machine. A good choice would be to use 8 or 16 MB. This may increase backup speed up to 5% compared to the speed achieved with the default fifo size.

−no_fifo
Don't use a *fifo* to optimize data flow from/to *tarfile*. Currently the −*fifo* option is used as default. (This may be changed at compile time.)

−shm Use System V shared memory for fifo. Normally **star** is compiled to use mapped /dev/zero pages for the fifo, if the operating system supports this. If **star** is compiled to have both code for mapped pages and for System V shared memory, **star** will use shared memory instead of the default. If the *−help* menu doesn't show the *−shm* flag you have no choice. When using System V shared memory, you may have to raise the system's internal limit for shared memory resourses to get enough shared memory for **star**.

−v Be verbose. This normally results in more output during operation. See also in the description for the *−t* flag. Normally, **star** does its work silently.

−tpath Use this option together with the *−t* option to get only a list of the pathnames of the files in the archive. This may be used in shell scripts to generate a name list. If used together with the *−diff* option, **star** will only print the names of the files that differ. A second run of **star** may then be used to restore all files that had differences to the archive. Use the *list=* option to specify the namelist in this case.

H=headertype

Generate a tape archive in *headertype* format. If this option is used in extract/list mode this forces **star** to interpret the headers to be of type *headertype.* Valid parameter for *headertype* are:

help Print a help message about possible headertypes.

tar Old UNIX tar format. This archive format may only store plain files, directories and symbolic links. Pathnames longer than 99 chars may not be archived. See also the −d option as a note to some implementations.

star Old **star** standard format. This is an upward/downward compatible enhancement of the *old* UNIX tar format. It has been introduced in 1985 and therefore is not Posix compliant. **Star** format allows to archive special files (even sockets) and records access time and creation time. Newer versions of the old **star** format allow very long filenames (> 100+155 chars) and sparse files. This format is able to copy the non Posix compliant device nodes on HP-UX that have 24 bits in the minor device number, which is more then the maximum allowed which is 21 bits.

gnutar This is a commonly used, not Posix compliant enhancement to the old tar format. The implementation of the gnutar archive format within **star** is not complete, but sufficient for most gnutar archives. See NOTES for more information.

ustar IEEE/Posix1003/IEC-9945-1 Standard Data Interchange format. With this option in effect, **star** will generate 100% ansi compliant tar archives. Files with pathnames longer than 100+155 chars may not be archived. If **star** is called as **ustar** the default archive format is **ustar**.

xstar Extended standard tar format. **Star** uses the **xstar** format as default archive format. This is an upward/downward compatible enhancement of the IEEE/Posix1003/IEC-9945-1 Standard Data Interchange format. It allows among others very long filenames (> 100+155 chars) and records access time and creation time.

All tar archive formats may be interchanged if the archive contains no files that may not be archived by using the old tar format. Archives in the *xstar* format may be extracted by any 100% ansi compliant tar implementation if they contain no files with pathnames >100+130 chars and if they contain no sparse files that have been archived by using the −sparse option.

C=dir Perform a **chdir**(2) operation to *dir* before storing next files. In all cases, **star** will perform the **chdir**(2) operation relative to the current working directory of the shell.

- In list mode (with the **−t** flag), **star** ignores all *−C* options.
- In create mode (with the **−c**, **−r** and **−u** flag), **star** walks through all *−C* options and file type arguments. While **tar**(1) goes back to the current working directory after

storing one file argument that immediately follows the −*C* option, **star** changes the directory only if a new −**C** option follows. To emulate the behavior of **tar**(1), add a −**C** . option after the file argument.

- In extract mode (with the −**x**, −**n** and −**diff** flag), **star** builds a pattern list together with corresponding directories and performs a **chdir**(2) to the corresponding directory of a matching pattern. All **pat=** options in this case are interpreted as if they were preceded by a −**C** . option. See EXAMPLES for more information.

−z run the input or output through a **gzip** pipe. This is currently a quick and dirty hack, that mainly will cover the most common usage to compress the tar output if it is a file. No reblocking will be done, so this option will currently only make sense on plain files. The environment variable **STAR_COMPRESS_FLAG** may be used to specify one option for gzip. If you want to write write compressed archives to tape, you should use
star -c . **|** *gzip* **|** *sdd ibs=4k obs=32k -fill of=/dev/nrst0*
or
star -c . **|** *gzip* **|** *sdd ibs=4k obs=32k -fill ovsize=60m of=/nrst0*
if the tape can hold 60 MB.

−B Force **star** to perform multiple reads (if necessary) to fill a block. This option exists so that **star** can work across the Ethernet, since pipes and sockets return partial blocks even when more data is coming. If **star** uses *stdin* as archive file, **star** behaves as if it has been called with the −**B** option.

−i Ignore checksum errors on tar headers. If this option is specified, **star** will not exit if a header with a bad checksum is found but search for the next valid header.

−d Do not store/create directories. Old versions of **tar** such as published with the seventh edition of UNIX are not able to deal with directories in tar archives. If a tar archive is generated without directories this avoids problems with tar implementations found on SYSVr3 and earlier.

−m Do not restore access an modification time. (Access time is only available if **star** is reading *star* or *xstar* archives.) If **star** extracts other archive types, the −*m* flag only referres to the modification time.

−nochown
Do not restore owner and group of files. This may be used if super user privileges are needed to overwrite existing files but the local ownership of the existing files should not change.

−atime, **−a**
Reset access time of files after storing them to *tarfile.* On Solaris 2.x, (if invoked by root) **star** uses the *_FIOSATIME* ioctl to do this. This enables **star** not to trash the *ctime* while resetting the *atime* of the files. If the −**atime** option is used in conjunction with the list command, **star** lists access time instead of modification time. (This works only in conjunction with **star** and **xstar** format.)

−p Restore filemodes of directories. Without this option directories are created using the present **umask**(2). If in create mode i.e. storing files to archive, **star** stores directories past the corresponding files. This guarantees that even old tar implementations will be able to restore the correct times of directories.

−l Do not print a warning message if not all links to hard linked files could be dumped. This option is evaluated in the opposite way to **tar**(1).

−L Follow symbolic links as if they were files. Normally **star** will not follow symbolic links but stores their values in *tarfile.*

−D Do not descend directories. Normally, **star** descends the whole tree if it encounters a directory in in its file parameters.

−M Do not descend mount points. This is useful when doing backups of complete filesystems. See

NOTES for more information.

−I Do interactive creation, extraction or renaming. For every file that matches the list of patterns and that has a more recent modification time in the tar archive (if in extract mode and the −U option is not specified) **star** prints its name and asks:

get/put ? Y(es)/N(o)/C(hange name) :

You may answer either ‘N’ for No or <Return> to skip this file. If you answer ‘Y’ the file is extracted or archived on tape with its original name. If you answer ‘C’, you are prompted for a new name. This name is used for the filename on disk if **star** is in extract mode or for the archive name if **star** is in create mode.

−O Be compatible to old versions of tar. If **star** is invoked with this option, **star** generates archives which are fully compatible with old UNIX tar archives. If in extract mode, **star** ignores any additional info in the headers. This implies neither that archives generated with this option are binary equal with archives generated by old tar versions nor that **star** is trying to comprehend all bugs that are found in old tar versions. The bug in old tar versions that cause a reversal of a space and a NULL byte in the checksum field is not repeated. If you want to have signed checksums you have to specify the −singed_checksum option too. This option is superseeded by the **H=**headertype option.

−P Allow **star** to write a partial record as the last record. Normally, **star** writes each record with the same size. This option is useful on unblocked tapes i.e. cartridge tapes like QIC tapes as well as with archives that are located in files.

−S Do not store/create special files. You need to be super user to extract special files.

−U Restore files unconditionally. By default, an older file will not replace a corresponding newer file on disk.

diffopts=optlst
Comma separated list of diffopts. Valid members in *optlst* are:

help Print a summary of possible members of the diffopts list.

not Invert the meaning of all members in the diffopts list i.e. exclude all present options from the compare list.

perm Compare file permissions. With this option in effect, **star** compares the low order 12 bits of the st_mode field.

mode Same as *perm.*

type Compare file type.

nlink Compare link count on hardlinks (currently not supported).

uid Compare numerical user id of file.

gid Compare numerical group id of file.

uname Compare ASCII version of user id of file.

gname Compare ASCII version of group id of file.

id Compare all user/group related info of file.

size Compare file size.

data Compare content of file.

cont Same as *data.*

rdev Compare major/minor numbers for device nodes.

hardlink Compare target of hardlinks.

symlink Compare target of symlinks. This evaluates the value returned by the readlink(2) call.

atime Compare access time of file. This only works with tar archives in *star* and *xstar* format.

mtime Compare modification time of file.

ctime Compare creation time of file. This only works with tar archives in *star* and *xstar* format.

times Shorthand for: *atime,mtime,ctime*.

If *optlst* starts with a ! the meaning of all members in *optlst* is inverted as with the *not* optlist member.

If *diffopts* are not specified, **star** compares everything but the access time of the files.

pattern=pattern, **pat=**pattern

Set matching pattern to *pattern.* A maximum of 100 pattern=pat options may be specified. If more than one pattern is specified, a file matches if any of the specified pattern matches. Patterns may be used in create mode to select or exclude files from the list of file type arguments or the files located in a sub tree of a file type argument directory. In extract or list mode, all file type arguments are interpreted to be select or exclude patterns. Note that the method to restore subtrees used by **tar**(1) does not work with **star**. To extract a complete sub tree from the directory *dir* with **star** use *dir/** instead of simply *dir/* (see manual page for match(1) for more details of the pattern matcher). All patterns are selection patterns by default. To make them exclude patterns, use the **-V** option.

−not, **−V**

Invert the meaning of the pattern list. i.e. use those files which do not match any of the pattern.

list=filename

Read filenames for store/create/list command from *filename.* The file *filename* must contain a list of filenames, each on a separate line. This option implies the −D option.

VOLHDR=name

Use *name* to generate a volume header.

−xdir Extract directories even if the corresponding directories on the archive are not newer. This is useful when for some reason, the directories are recorded after their content, or when the permissions of some directories must be set in any case.

−keep_old_files, **−k**

Keep existing files rather than restoring them from *tarfile.* This saves files from being clobbered even if *tarfile* contains a more recent version of the corresponding file.

−/ Don’t strip leading slashes from file names. Tar archives containing absolute pathnames are usually a bad idea. With other **tar** implementations, they may possibly never extracted without clobbering existing files. **Star** for that reason, by default strips leading slashes from filenames.

maxsize=#

Do not store files in *tarfile* if they are bigger than # kBytes.

newer=filename

Do not store files to *tarfile* if their modification time is not newer than the modification time of *filename.* See **−ctime** option for changing this behavior.

−ctime If used with the list command, this lists *ctime* rather than *mtime.* If used with the extract command, this tries to restore even the *ctime* of a file by generating time storms. You should not do this when in multi user mode because this may confuse programs like cron and the news system. If used with the create command this changes the result of the *newer=* option. **Star**, in this case compares the *ctime* of all files to the *mtime* of the stamp file rather then comparing the *mtimes* of both files.

bs=# Set output block size to #. You may use the same method as in **dd**(1) and **sdd**(1). The number representing the size is taken in bytes unless otherwise specified. If a number is followed directly by the letter ‘b’, ‘k’ or ‘m’, the size is multiplied by 512, 1024 or 1024∗1024. If the size consists of numbers separated by ‘x’ or ‘∗’, multiplication of the two numbers is performed. Thus *bs=7x8k* will specify a blocksize of 56 kBytes. Blocksize must be a multiple of 512 bytes. See also the description of the obsolete *blocks=* option for more details on blocksizes.

fs=# Set fifo size to #. See *bs=* for possible syntax. The default size of the fifo is 1 Mbyte. See **−fifo** option for hints on using the right fifo size.

tsize=# Set tape volume size to # 512 byte blocks. With this option in effect, **star** is able to archive filesystems that are bigger then the tape size. Files that do not fit on a single tape may not be stored with the current version of **star**.

−qic24 Set tape volume size to 61440 kBytes.

−qic120
Set tape volume size to 128000 kBytes.

−qic150
Set tape volume size to 153600 kBytes.

−qic250
Set tape volume size to 256000 kBytes.

−nowarn
Do not print warning messages. This sometimes is useful to make the output more readable.

−time Print timing info. See DIAGNOSTICS for more information.

−no_statistics
Do not print statistic messages at the end of a **star** run.

−fifostats
Print fifo statistics at the end.

−numeric
Use the numeric user/group fields in the listing rather than the default. The default is to list the ASCII version of user/group of the file.

−newest
In conjunction with the list command this lists you only the newest file in *tarfile.*

−newest_file
In conjunction with the list command this lists you only the newest regular file in *tarfile.*

−signed_checksum
Use signed chars to calculate checksums. This violates the tar specs but old versions of **tar** derived from the seventh edition of UNIX are implemented in this way. Note: Only filenames and linknames containing chars with the most significant bit set may trigger this problem because all other fields only contain 7 bit ASCII characters, octal digits or binary zeroes.

−sparse
Handle files with holes effectively on store/create.

−force_hole
Try to extract all files with holes. This even works with files that are created without the **−sparse** option. **Star**, in this case examines the content of the files in the archive and replaces writes to parts containing binary zeroes with seeks. This option should be used with extreme care because you sometimes get in trouble when files get unattended holes.

−to_stdout
Extract files to stdout. This option may be used to extract tarfiles containing tarfiles (see examples below).

−wready
This option is added as a hack for a bug in the SunOS/Solaris *st* device driver. This driver has problems to sense the loading time with Exabyte drives with factory settings. **Star** waits up to one minute for the drive to become ready if this option is specified.

−force_remove
Force to remove non writable files on extraction. By default, **star** will not overwrite files that are read only. If this option is in effect, **star** will silently remove these files to allow the extraction of a file.

−ask_remove
Ask to remove non writable files on extraction. By default, **star** will not overwrite files that are read only. If this option is in effect, **star** will ask whether it should remove these files to allow the extraction of a file in the following way:

remove '*filename*' ? Y(es)/N(o) :

−remove_first
Remove files before extraction. If this option is in effect, **star** will remove files before extracting a file from the archive. This is needed if you want to change the file type or if you need to break a hard link. If you do not use either **−ask_remove** or **−force_remove** together with **−remove_first**, this option is useless and no files will be removed.

−remove_recursive
Remove files recursive. If removing of a file is permitted, **star** will only remove files, specials and empty directories. If this option is in effect, **star** will be allowed to recursively removes non empty directories too.

−nullout
Do not actually write to the archive but compute and add the sizes. This is useful when trying to figure out if a tape may hold the current backup.

−debug Print debug messages. Among other things, this gives debug messages for headertype recognition, tar type properties, EOF recognition, opening of remote archives and fifo internals.

SIGNALS

If **star** handles a signal, it first prints the statistics. **Star** handles the following signals:

SIGINT usually generated by ^C from the controlling tty. Upon receipt of a SIGINT, **star** prints statistics and exits. If in create mode i.e. storing files to archive, **star** finishes with the current file to ensure that no partial file is written to the archive, write an eof record and then exits.

SIGHUP not to be generated from a tty. The actions are the same as upon receipt of a SIGINT.

SIGQUIT usually generated by ^\ from the controlling tty. Upon receipt of a SIGQUIT, **star** prints statistics and continues with the current operation. This is useful to watch the progress of the current operation.

EXAMPLES

To get a listing in a way similar to ls −l one might use:

example% star −tv f=/dev/nrst1

To copy the directory tree in */home/jes* to the directory */home/fs* use:

example% (cd /home/jes; star −c .) | (cd /home/fs ; star −xp)

or by using the change directory options of **star**:

example% star −c -C /home/jes . | star −xp -C /home/fs

To compare the content of a tape to the filesystem one might use:

example% star −diff −v f=/dev/nrst1

To compare two directory trees one might use:

example% star −c . | (cd todir ; star −diff −v diffopts=!times)

To extract a backup of the /usr tree without all files residing below /usr/openwin one might use:

example% star −xp −V pat=openwin/\∗ f=/dev/nrst1

To extract all *.c* files to **src**, all *.o* files to **obj** and all other files to /tmp one might use:

example% star −xp -C src '∗.c' -C obj '∗.o' -C /tmp '∗' f=/dev/nrst1

To extract a zipped tar file that is located on a read only filesystem e.g. a CD in /tmp while having the shell's working directory on the CD one might use:

example% star −zxp −C /tmp f=star-1.1.tar.gz

To backup a list of files generated by the **find**(1) command:

example% find . *find_options* **−print | star −c list=− f=/dev/nrst1**

To extract tarfiles that contain tarfiles one might use:

example% star −x −to_stdout f=/dev/nrst1 pat=*pat* **| star −xp**

Pat, in this case should match the tarfile in the tarfile on tape that should be extracted.

To make a backup of the root filesystem to a tape drive connected to a remote machine, one might use:

example# cd /
example# star −cM bs=63k f=tape@remotehost:/dev/nrst1 .

You need a line in /etc/passwd like the following to enable this:

tape:NP:60001:60001:Tape:/etc/tapehome:/etc/rmt

And a .rhosts file in /etc/tapehome to allow remote connections from the appropriate hosts.

To repair a corrupted filesystem for which no recent backup exists, do the following:

example# fsck −y /filesys
example# mount /filesys
example# cd /filesys
example# star −xpk f=/dev/nrst1
example# mt −f /dev/nrst1 rewind
example# star −diff −v diffopts=!times f=/dev/nrst1

Now check the differences and decide whether to restore additional files. This may be done by generating a list containing the needed filenames and using the list= option or by using the interactive mode (see −I option).

If you want a list that only contains all filenames from files with differences you may use:

example# star −diff −tpath diffopts=!times f=/dev/nrst1

If you are looking for files that changed the type or the access permission because this is a common case on still corrupted files, use:

example# star −diff −tpath diffopts=type,perm f=/dev/nrst1

FILES

None.

SEE ALSO

tar(1), **cpio**(1), **rcp**(1), **mt**(1), **rmt(1), match**(1), **dd**(1), **sdd**(1), **star**(4/5), **rcmd**(3)

DIAGNOSTICS

star: f records + p bytes (total of x bytes = d.nnk).

The number of full records, the number of bytes in partial records and the total amount of data in KBytes.

star: Total time x.yyysec (z kBytes/sec)

The time used and the transfer speed from/to the archive.

NOTES

Star strips leading ./ sequences from pathnames. This lets **star** in many cases store longer pathnames than other implementations.

The ansi method (ustar format) of storing files with pathnames that are longer than 100 chars has some limitations:

> The name field (100 chars) an inserted slash (‘/’) and the prefix field (155 chars) produce the pathname of the file. When recreating the original filename, name and prefix are concatenated, using a slash character in the middle. If a pathname does not fit in the space provided or may not be split at a slash character so that the parts will fit into 100 + 155 chars, the file may not be archived. Linknames longer than 100 chars may not be archived too.

The *star, xstar* and *gnutar* archive formats don’t have these limitations. While gnutar uses a method that makes it impossible for other tar implementations (except **star**) to restore filenames that are longer than 100 chars, the *xstar* archive format uses a method that allows an ansi compliant way of storing filenames, if the ansi method would allow this.

Some buggy tar implementations will generate incorrect filenames during a restore operation if the archive contains pathnames of exactly 100 chars length.

Star adds a *tar* signature in the last four bytes of each tar header. This is no problem with the **star** archive format. On the other side, the **xstar** archive format claims to be as Posix compliant as possible. Inserting this *tar* signature is a minor deviation from the standard that has the last 12 bytes of each header reserved for future use. On the other side, tar implementations such as **pax** that only compute checksums on the first 500 bytes of the header are violating the standard. All tar implementations that are 100% Posix compliant will be able to extract **xstar** archives as long as no new standard is defined that claims the last 12 bytes of the header for a different use. But then the version number should be changed from ‘00’ to ‘01’.

Star uses these four bytes since 1985 without problems. If for some reason a new standard will use the reserved bytes, it will be easy to change the **xstar** format so that it will be 100% Posix compliant. The probability of falsely detecting other tar formats as **xstar** format however will be higher in this case.

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

The way EOF is handled differs, weather the fifo is in effect or not. If the fifo is not used, **star** stops reading the archive if it encounters a logical EOF record in the archive. If the fifo is used, **star** reads until the real EOF mark on tape is reached.

Gnu tar often creates tar archives with incorrect logical EOF marks. The standard requires two blocks that are completely zeroed, whereas gnutar often only adds one of them.

Old versions of tar found on SYSVr3 and eralier cannot read tar archives with a blocksize greater than 10 kBytes.

The method of storing *sparse* files currently used with the *star* and *xstar* format is not guaranteed to be used in later versions of **star**. If the author decides to change this method, later versions of **star** may not be able to restore sparse files from tar archives made by the current version of **star**.

Some tar implementations violate the standard in using only the first 500 Bytes of the header for checksum computation. These tar implementations will not accept *star* and *xstar* type tar archives.

Sun's Solaris 2.x tar implementation violates the Posix standard. Tar archives generated by **star** cause Sun's tar to print tar: impossible file type messages. You may ignore these messages.

Gnutar's dumpdirs are currently not implemented.

If gnutar archives sparse files with more than four holes, it produces archives that violate the standard in a way that prevents other tar implementations to read these archives. **Star** knows about that and is able to handle these gnutar archives.

The filetype *N* (LF_NAMES) from gnutar (an obsolete method of storing long names) will never be implemented.

SUID NOTES

If **star** is installed suid root, **star** is able to make connections to remote archives for non root users. This is done by using the rcmd(3) interface to get a connection to a rmt(1) server.

Star resets its effective uid back to the real user id immediately after setting up the remote connection to the rmt server.

LIMITATIONS

Star currently handles files up to a size of 2 GB and archives up to 2000000 TB. The maximum file size limitation of 2 GB may easily be expanded to 8 GB. With a non standard extension **star** will handle files up to 200000000 TB.

Access control list are currently not handled.

BUGS

There is currently no way to set the fifo lowwater and highwater marks.

There is currently no way to automatically delete files in the target file tree if they are obsolete. **Star** should implement something similar to gnutar's dumpdirs.

If not invoked by the super user **star** may not be able to extract files if they reside in read only directories.

Star is not able to make a complete backup of a filesystem if files are hidden by a mount that is in effect on a directory of this filesystem. This could be avoided if the loopback filesystem had an option that tells *lofs* not to traverse mountpoints.

The actual version of the IEC-9945-1 standard has been changed to allow character specials, block specials and fifo files to have garbage sizes in the tar archive header. The current version of **star** does not implement this. This implies that tape archives that use this feature may not be processed by **star**.

On Solaris systems, **star** should use the _FIOAI file ioctl to examine sparse files.

HISTORY

Star was first created in 1982 to extract tapes on a UNIX clone that had no tar command. In 1985 the first fully functional version has been released as **mtar.**

When the old **star** format extensions have been introduced in 1985, it was renamed to **star** (Schily tar). In 1994, Posix 1003.1 extensions were added and **star** was renamed to **star** (Standard tar).

AUTHOR

Jörg Schilling
Seestr. 110
D-13353 Berlin
Germany

Mail bugs and suggestions to:

joerg@schily.isdn.cs.tu-berlin.de or **js@cs.tu-berlin.de**