Sustainability of Digital Formats: Planning for Library of Congress Collections

Tape Archive (tar) File Format Family

>> Back

Table of Contents

Format Description Properties

- ID: fdd000531
- Short name: tar
- Content categories: aggregate
- Format Category: file-format
- Other facets: container-bundle, binary, structured
- Last significant FDD update: 2024-04-30
- Draft status: Full

Identification and description

Full name	Tape Archive (tar) File Format
Description	A tar (tape archive) file format is an archive created by tar, a UNIX-based utility used to package files together for backup or distribution purposes. It contains multiple files (also known as a tarball) stored in an uncompressed format along with metadata about the archive. Tar files are not compressed archive files. They are often compressed with file compression utilities such as gzip or bzip2. Each file object includes any file data, and is preceded by a 512-byte header record. The file data is written unaltered except that its length is rounded up to a multiple of 512 bytes. At the end of the archive file there are two 512-byte blocks filled with binary zeros as an end-of-file marker. The file header record contains metadata about a file. To ensure portability across different architectures with different byte orderings, the information in the header record is encoded in ASCII. Tar archives are fully compatible between UNIX and Windows systems because all header information is represented in ASCII. See Notes for more information about the capitalization of tar and Unix. The tar file format has changed over time as additional functionality has been developed for the tar UNIX utility leading to format extensions that include additional information for necessary implementations beginning in the 1980s. Early versions of tar formats were inconsistent in how numeric fields were constructed that were corrected in later implementations to improve portability of the format, beginning with the first POSIX standard for tar file formats in 1988.

	The POSIX.1 2001 introduced the "extended tar", tar.h, or pax format which added vendor-tagged or vendor-specific functionality. This is the most flexible format with the richest features of other tar archive specifications. As stated in gnu.org's documentation about various iterations of tar file formats, "This format is quite recent, so not all tar implementations are able to handle it properly. However, this format is designed in such a way that any tar implementation able to read 'ustar' archives will be able to read most "posix" archives as well." The POSIX.1 2001 specification relieved the file size of 8 GB of previous tar formats. The new tags as described in freebsd.org's tar documentation are as follows: • atime, ctime, mtime = File access, inode change (time when the file metadata was last changed), and modification times. • path = The full pathname of the entry. Note that this is encoded in UTF-8 and can thus include non-ASCII characters. • linkpath and symlink = The full path of the linked-to file. This can also include non-ASCII characters as it is encoded in UTF-8. • uname, gname = The user name and group name stored here are encoded in UTF-8 and can thus include non-ASCII characters. • size = The size of the file. Note that there is no length limit on this field, allowing conforming archives to store files much larger than the historic 8GB limit. • uid, gid = Similar to the user name and group name fields, UID (user ID) and GID (group ID) fields can be of arbitrary length. The POSIX.1 2001 standard also features changes to the applicable typefield values. This extended tar or tar.h archive format stores new data in ustar-compatible archive entries that use "x" or "g" typeflags. FreeBSD, an open source Unix-like operation system, provides documentation of tar file format versions and stresses the compatibility between extended tar formats and ustar tar archives defined in the POSIX.1 1988 standard. "older implementations that do not fully support these extensions will extract the metadata into regular files, where the metadata can be examined as necessary." The POSIX.1 2001 standard defined the pax utility and pax format that serves as an extension of the tar format. The pax utility uses "-x" in the command string to output the archive format as ustar. Opengroup.org's Pax documentation clarifies that the pax utility supports the ustar format, defined as, "The tar interchange format; see the EXTENDED DESCRIPTION section. The default blocksize for this format for character special archive files shall be 10240. Implementations shall support all blocksize values less than or equal to 32256 that are multiples of 512." The tar file format doesn't feature native data compression, so tar archives are often compressed with an external utility such as; gzip, bzip2, XZ (using 7-Zip / p7zip LZMA / LZMA2 compression algorithms), Brotli, Zstandard, and similar tools to reduce the archive's size for portability and data backup. Resulting compressed files can be found named with single extension, e.g. tgz, tbz, txz, tzst, or with double file extension, e.g. tar.gz, tar.br, tar.bz2, tar.xz, tar.zst For an overview of tar version history, See Notes.
Production phase	May be used at any life-cycle phase for bundling files. When compressed with an external software program, maybe used at any life-cycle phase for packaging files for exchange and portability.
Relationship to other formats	
Used by	gzip, GZIP. One common workflow is to use the tar utility to create an archive of files, their attributes, and their directory structure into a single .tar file, and then compress it with compress to make a .tar.Z file.

Local use

LC experience or existing holdings	The Library has over 5,000 tar files inventoried on long-term storage.
LC preference	The Library of Congress Recommended Formats Statement (RFS) lists ZIP as both a preferred (for direct file submission) and acceptable (on a mass storage device) format for packaged delivery of Software and Video Games.

Sustainability factors

Disclosure	The 2001 format specification for tar file formats is maintained by the IEEE (Institute of Electrical and Electronics Engineers) and is openly available.
Documentation	The 2018 POSIX standard, jointly developed and maintained by the Open Group and IEEE is publicly available at Open Group's site. See Format Specifications.
Adoption	Tar file formats are immensely popular on UNIX and UNIX-like systems due to the ease of use of tar commands. Tar files are frequently used in conjunction with external file-based compression schemas for portability and including functions such as encryption and integrity checks. The chosen compression schema influences compression ratios and speeds, competing with ZIP, RAR, and other archive formats. The following is a non-exhaustive list of software applications that open tar files: Windows • File Viewer Plus • Smith Micro StuffIt Deluxe • Corel WinZip • RARlab WINRAR • PeaZip • GNU tar • Estsoft Alzip • 7-zip • Zipeg Mac OS • Apple Archive Utility • Corel WinZip Mac • Smith Micro StuffIt Deluxe Mac 116 • Incredible Bee Archiver • GNU tar • Zipeg • Keka Linux • GNU tar Comments welcome.
Licensing and patents	None.
Transparency	Traditional tar files are uncompressed so individual items can easily be extracted. Transparency of compressed tar files are dependent upon algorithms and tools used to read the file. Easily compatible with UNIX and Windows systems as all file header information is represented in ASCII.
Self-documentation	The tar format provides no metadata support beyond what is needed to support unpacking the archive and extracting the component items into a file system. **Accessibility Features**

	No specific features in the file format. Features to support accessibility would be found in the bundled and compressed files (such as embedded captions and subtitles in audiovisual content, tagged and structured text in textual documents, and alt text for images). Aggregate files can also contain separate files for transcripts, timed text or captions as part of the bundled package. See Relationships to other formats for details.
External dependencies	None. Creating tar files can be done via command line in both UNIX or Linux systems as well as a graphic interface in software such as 7z. No external dependencies beyond available software to extract and decompress a compressed tar file.
Technical protection considerations	Tar files do not natively support encryption but its possible to encrypt compressed tar files with external software programs.

Quality and functionality factors

Aggregate	
Compression	Tar files do not feature native compression but instead contain uncompressed byte streams of files. There are a wide variety of compression programs that can compress tar files including; gzip, bzip2, and many others.

File type signifiers and format identifiers

Tag	Value	Note
Filename extension	tar	
Internet Media Type	application/x-tar	There is no registration at IANA for an Internet Media Type for the tar format. The application/x-tar value can be found at File-Extensions.org
Internet Media Type	application/x-gtar multipart/x-tar application/x-compress application/x-compressed	Several different Internet Media Types are in use for compressed tar files such as .tar.gz, .tar.bz2, and .tar.z. See File-Extensions.org and Wikipedia entry for list of archive formats.
Magic numbers	Hex: 75 73 74 61 72 ASCII: ustar	Magic numbers for an uncompressed POSIX ustar file [257 (0x101) byte offset] from the 2001 IEEE standard. From garykessler.net.
Magic numbers	Hex: 42 5A 68	Magic numbers for a tar files compressed with bzip2. See garykessler.net.
Magic numbers	Hex: 1F 9D	Magic numbers for tar.z file, compressed tape archive file using standard LZW (Lempel-Ziv-Welch) compression. See garykessler.net.
Magic numbers	Hex: 1F A0	Magic numbers for tar.z file, compressed tape archive file using LZH (Lempel-Ziv-Huffman) compression. See garykessler.net.
Magic numbers	Hex: 1F 8B	Magic numbers for TAR.GZ file, compressed tape archive file using GZIP. See Wikipedia entry for list of file signatures.
Magic numbers	Hex: FD 37 7A 58 5A 00	Magic numbers for any file format compressed with the XZ compression utility including tar.xz files. See Wikipedia entry for list of file signatures and XZ at fileformats.archiveteam.org.
Uniform Type Identifier (Mac OS)	public.tar-archive	Apple Uniform Type Identifier. See https://www.nationalarchives.gov.uk/pronom/x-fmt/265. Outline record only.
Other	NF00423	See https://www.archives.gov/files/lod/dpframework/id/NF00423.ttl.
Pronom PUID	x-fmt/265	See http://www.nationalarchives.gov.uk/PRONOM/x-fmt/265.
Wikidata Title ID	Q283579	See https://www.wikidata.org/wiki/Q283579.

Notes

General	Tar can reference both the UNIX command to great the archive file format as well as the file itself, both with a lowercase spelling. The POSIX 2001.1 standard references the file format as the extended tar or "tar.h" file format while the IEEE 1988 Standard Interpretation defines the file format as "tar" in lowercase as well. For clarification purposes, when referencing the file format, this format description document will use "tar files" or the "tar file format." The term UNIX generally refers to the licensed operating systems developed in 1996 and trademarked by the Open Group (link via Internet Archive). The Linux Information Project helps to provide comprehensive information about Linux and other free software but specifically explains how UNIX is defined and appropriate capitalizations of the term. Throughout this document, upper case UNIX refers to the trademarked operating systems. As described in the Linux Information Project's description, "Unix-like" or "UNIX-like" "is commonly used as a generic term referring to all operating systems that incorporate the major features of the early versions of UNIX, whether or not they officially call themselves UNIX or use the UNIX trademark. It is a broader term than Unix in the sense that the addition of the word -like eliminates any claim or implication that any system is UNIX (regardless of how UNIX might be defined, or spelled), and instead merely indicates that a system resembles the original UNIX systems. Thus, it is better at avoiding the controversial issues regarding what is, or can legally be called, UNIX, or Unix." Comments welcome.
History	The tar file format was first introduced in 1979, with Version 7 UNIX, as the tar utility was used to write data to tape drives. These tape drives were data storage devices that would read and write data on magnetic tape. These older tar archive format headers consisted of 10 elements. The bracketed numbers in the list below represent the number of bytes allowed in each field. All unused bytes in the header record are filled with nulls. • char-name[100] = name of file • char-mode[8] = file mode, stored as an octal number in ASCII • char-uid[8] = owner user ID • char-gid[8] = owner group ID • char-size[12] = length of file in bytes • char-mtime[12] = modify time of file • char-checksum[8] = checksum for header • char-linkflag[1] = indicator for links • char-linkname[100] = name of linked file • char-pad[255] = character padding Early tar formats contained various inconsistencies within numeric fields. Early implementations filled numeric fields with leading spaces, which was corrected by the IEEE (POSIX.1) 1003.1-1988 standard where numeric fields were filled with leading zeroes for better portability. The tar archive file format was officially standardized by the POSIX 1988 standard, creating the UNIX Standard tar or "USTAR" format. The POSIX.1 2001 standard introduced additional header fields which provide more information about the file and its archived contents. According to Wikipedia, "The ustar format allows for longer filenames...the maximum filename size is 256, but it is split among a preceding path "filename prefix" and the filename itself, so can be much less." The POSIX 1988 standard tar utility can determine a USTAR format's presence based on the string "ustar" in the magic field. POSIX 1988 tar file headers contain additional elements than pre-POSIX file headers including: • char-typeflag[1] = type of entry.

- char-magic[6] = USTAR indicator. Contains the magic value "ustar" which is then followed by a null byte to indicate a POSIX standard.
- char-version[2] = USTAR version. This should be two copies of the ASCII digit zero for the POSIX standard archive.
- char-uname[32] = owner user name.
- char-gname[32] = owner group name.
- char-devmajor[8] = device major number. Major and minor numbers for character device or block device entry.
- char-devminor[8] = device minor number. Major and minor numbers for character device or block device entry.
- char-prefix[155] = prefix for filename. This field provides an opportunity to input information about the pathname if its too long for the allotted 100 bytes. If the prefix field is not empty, the reader will prepend the prefix alue and a / character to the name field to create the full pathname.

This "typeflag" field serves as an extension of the older "link" field in older tar formats. Typeflag field values are listed as follows and can be found illustrated in PTC MKS Toolkit's tar utility:

- 0 or null = Regular file
- 1 = Link to another file already archived
- 2 = Symbolic link
- 3 = Character special device
- 4 = Block special device
- 5 = Directory
- 6 = FIFO special file
- 7 = Reserved
- A-Z = Available for custom usage

The POSIX.1 2001 standard introduced the "extended tar", tar.h, or pax format which added vendor-tagged or vendor-specific functionality. This is the most flexible format with the richest features of other tar archive specifications. A thorough explanation of the POSIX.1 2001 standard and the tar.h format can be found in the Identification and Description section above.

GNU, a series of open-source software programs has it's own implementation of the tar utility (from versions 1.13.25) to create tar files dating to pre-POSIX tar formats, adding improvements such as incremental archives. According to GNU's comparison of tar iterations (link via Internet Archive) these features that were implemented make this tar format incompatible with other archive formats. GNU tar has the ability to read POSIX.1 2001 standard tar files.

For more robust definitions of POSIX fields, see Identification and Description.

Format specifications

- File Format Specifications
  - IEEE 1003.1-1988 - IEEE Standard Portable Operating System Interface for Computer Environments (https://standards.ieee.org/standard/1003_1-1988.html). Available by paid subscription only.
  - 1003.1-2001 - IEEE Standard for IEEE Information Technology - Portable Operating System Interface (POSIX(TM)) (https://ieeexplore.ieee.org/document/974398). Specifications for tar.h file format headers can be found on page 376 and 377.
  - 1003.1-2008 - IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(R)) (https://ieeexplore.ieee.org/document/4694976). Specifications for tar.h file format headers can be found on page 409 and 410.
  - 9945-2009 - IEEE/ISO/IEC International Standard - Information technology Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7 (https://ieeexplore.ieee.org/document/5393893). Specifications for tar.h file format headers can be found on page 409 and 410.
  - The Open Group Base Specifications Issue 7, 2018 edition IEEE Std 1003.1-2017 (https://pubs.opengroup.org/onlinepubs/9699919799/basedefs/contents.html).

Useful references

URLs

- Format Specification Interpretations
  - IEEE 1003.1-1990 - Information Technology -- Portable Operating System Interface (POSIX(TM)) -- Part 1: System Application Program Interface (API) [C Language] (https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/interpretations/1003.1-1990_interp.zip). Interpretations provide the IEEE an opportunity to prepare responses to regarding the application of IEEE standards.
  - IEEE 1003.1-1988 - Standards Interpretations for IEEE Standard Portable Operating System Interface for Computer Environments (https://ieeexplore.ieee.org/document/182902). Interpretations provide the IEEE an opportunity to prepare responses to regarding the application of IEEE standards.
- Compression of tar files.
  - Using Less Space through Compression (link via Internet Archive (https://web.archive.org/web/20201205032913/https://www.gnu.org/software/tar/manual/html_section/tar_68.html).
  - How to Compress and Extract Files Using the tar Command on Linux (https://www.howtogeek.com/248780/how-to-compress-and-extract-files-using-the-tar-command-on-linux/).
  - GNU Gzip (https://web.archive.org/web/20201112042947/https://www.gnu.org/software/gzip/manual/gzip.html). GNU gzip manual which covers the compression of tar files. (Link via Internet Archive
  - XZ (http://fileformats.archiveteam.org/wiki/XZ). Fileformats.archiveteam.org's description of the XZ compression utility.
- Command Line Operations to Create tar Files.
  - MKS Toolkit Backup and Tape Handling Solutions Guide. (https://www.mkssoftware.com/docs/man1/tar.1.asp).
  - GNU Index of Command Line Options (https://www.gnu.org/software/tar/manual/html_node/Index-of-Command-Line-Options.html).
  - How to Extract Files From a .tar.gz or .tar.bz2 File on Linux. (https://www.howtogeek.com/409742/how-to-extract-files-from-a-.tar.gz-or-.tar.bz2-file-on-linux/).
  - Tar in Linux – Tar GZ, Tar File, Tar Directory, and Tar Compress Command Examples (https://www.freecodecamp.org/news/tar-in-linux-example-tar-gz-tar-file-and-tar-directory-and-tar-compress-commands/).
  - Creating A tar File in Linux Via Command Line Options. (https://www.cyberciti.biz/faq/creating-a-tar-file-linux-command-line/).
  - Read and write tar archive files. (https://docs.python.org/3/library/tarfile.html). Python programming documentation to read and write tar files.
- Wikipedia References and Sources
  - Wikipedia entry for Tar (computing). (https://en.wikipedia.org/wiki/Tar_(computing)).
  - Wikipedia entry for List of archive formats. (https://en.wikipedia.org/wiki/List_of_archive_formats).
  - Wikipedia entry for bzip2. (https://en.wikipedia.org/wiki/bzip2).
  - Wikipedia entry for the Pax command. (https://en.wikipedia.org/wiki/Pax_(command)).
- General References and Sources
  - Single UNIX Specification, Version 2. (https://pubs.opengroup.org/onlinepubs/7908799/xcu/tar.html). 1997 UNIX specification for tar commands and tar file formats.
  - UNIX standard Archive format, Tape Archive. (https://www.file-extensions.org/tar-file-extension).
  - Gzip compressed tar archive. (https://www.file-extensions.org/tgz-file-extension).
  - GCK'S FILE SIGNATURES TABLE. (https://web.archive.org/web/20221112073316/https://www.garykessler.net/library/file_sigs.html).
  - format of tar archives. (https://www.mkssoftware.com/docs/man4/tar.4.asp).
  - tar file formats. (https://www.freebsd.org/cgi/man.cgi?tar(5)).
  - Controlling the Archive Format. (https://www.math.utah.edu/docs/info/tar_8.html).
  - Looking to open a tar file? (https://www.corel.com/en/file-formats/tar-file/).
  - .TAR File Extension. (https://fileinfo.com/extension/tar).
  - Differences Between the Many Archive Compressed File Formats. (https://helpdeskgeek.com/free-tools-review/differences-between-the-many-archive-compressed-file-formats/). Comparison of archive formats.
  - Archive, compression comparison: 7Z, Brotli, RAR, ZIP, Zstandard... (https://peazip.github.io/archive-file-formats-comparison.html). Archive file format compression comparison.
  - File Magic Numbers (https://gist.github.com/leommoore/f9e57ba2aa4bf197ebc5). Additional information on magic numbers for archive formats.
  - Gzip Intro page. (https://www.gzip.org/). Gzip data compression homepage and documentation resource.
  - UNIX Definition. (http://www.linfo.org/unix_upper.html). Linux Information Project explanation of the term unix and its usage.
- NARA File Format Preservation Plan ID entry for NF00423 (https://www.archives.gov/files/lod/dpframework/id/NF00423.ttl). Information in NARA File Format Preservation Plan ID about tar.
- Tar file format entry at PRONOM (https://www.nationalarchives.gov.uk/pronom/x-fmt/265). PUID is x-fmt/265

- Wikidata entry for tar Q283579 (https://www.wikidata.org/wiki/Q283579). Outline record only.

Last Updated: 05/17/2024

Digital Preservation Home | Digital Formats Home