

UnTAR

Magnus Holmgren

COLLABORATORS

	<i>TITLE :</i> UnTAR		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Magnus Holmgren	April 12, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	UnTAR	1
1.1	UnTAR.guide	1
1.2	UnTAR.guide/Introduction	1
1.3	UnTAR.guide/Why UnTAR?	1
1.4	UnTAR.guide/Features	2
1.5	UnTAR.guide/Usage	2
1.6	UnTAR.guide/Limitations	3
1.7	UnTAR.guide/Acknowledgements	3
1.8	UnTAR.guide/Glossary	3
1.9	UnTAR.guide/Source	4
1.10	UnTAR.guide/Author	4

Chapter 1

UnTAR

1.1 UnTAR.guide

UnTAR 37.0

Public domain in 1998 by Magnus Holmgren

[Introduction](#)

[Why UnTAR? Features](#)

[Usage Limitations](#)

[Acknowledgements Glossary](#)

[Source Author](#)

1.2 UnTAR.guide/Introduction

Introduction

UnTAR is a small tool to unpack so called **tar** archives. Such archives often have the extension `.tar`, `.tgz`, `.taz`, or `.tar.gz`. They are created by `tar`, and optionally compressed by `gzip` (indicated by the "gz" or "z" in the suffix).

Tar files can be seen as the Unix equivalent of lha or zip files: the purpose is to store a number of files in a single file. By compressing the entire tar file, good compression ratios can be achieved, but archive processing is a bit clumsy. All operations require the entire archive to be decompressed and possibly recompressed later.

The reason for writing UnTAR is that the programs `tar` and `gzip` are fairly large, the Amiga versions keep the unix command line style and often require `ixemul`. UnTAR is written to be simple, compact, reasonably fast, and use an Amiga-like command line style.

1.3 UnTAR.guide/Why UnTAR?

Why UnTAR?

There are already several other programs with purposes similar to UnTAR, so why write yet another one? Well, it all started because Oliver Wagner wrote a program called UnTGZ (not related to UnTGZ by Andreas Kleinert). However, it had a few problems. The latest version I'm aware of sometimes generated empty files when it should create files with contents. I wrote to Mr. Wagner about it, and suggested a few improvements. However, I didn't get any reply, nor did I see any updates. I considered making my own version, but didn't know much about **tar** files, so nothing happened.

Time passed. Then I saw the sources to a program called UnTGZ in the **zlib** distribution archive. It was a small and simple .tgz-file unarchiver and lister. I decided to make an Amiga version; it turned out to be more a rewrite for the Amiga than a simple port (like Andreas Kleinert's port), and I added some new features.

When UnTAR was almost finished, I looked at Aminet to see if there was any program called UnTAR there already. I found a program called UnTAR and several similar programs. I still felt that my program offered some features the others didn't, so I decided to release it anyway. There were already two programs called UnTGZ, so why not another UnTAR? ;)

1.4 UnTAR.guide/Features

Features

Small and simple to use program.

Modest system requirements. OS 2.04 is required for operation, OS 2.1+ features are used if available. About 100 kB of memory is needed to process an archive.

Optionally prompts if files are about to be overwritten.

Uses AsyncIO for speed.

Includes a version optimized for a 68020 CPU or better.

Can be made resident.

1.5 UnTAR.guide/Usage

Usage

```
UnTAR [FILE] <File> [TO <Drawer>] [LIST] [KEEP] [NOPROMPT] [QUIET] [CASE] [PATTERNS <Pattern> ...]
```

FILE Name of the **tar** archive to process. This archive can optionally be compressed by **gzip**. UnTAR will not try to append any suffix if the specified file couldn't be opened.

TO Specifies where the files in the archive should be extracted. This must refer to an existing drawer.

This option is ignored if **LIST** is specified.

LIST List the specified archive rather than extract the files in it. Note that in this list, **protection bits** and file names are shown in Unix format rather than Amiga format, to faithfully display the archive contents. The protection bits and file names are converted as needed when extracted.

The summary at the bottom contains three numbers: · Total size of the listed files. · Number of bytes gained through compression - if any - in percent for the archive as a whole, not just the listed files. · Number of files listed.

KEEP By default, UnTAR will delete any files that it was unable to extract completely. By specifying this option, partial files will not be deleted.

This option is ignored if **LIST** is specified.

NOPROMPT Do not show a prompt if a file to be extracted would overwrite an already existing file; they will be silently overwritten. The prompt accepts four answers: **y** = yes, overwrite the file; **n** = no, do not overwrite the file; **a** = all, overwrite the file and any further files; **s** = skip, do not overwrite the file nor any further files. Any answer not matching the above is considered to be **n**.

If there is no interactive input (i.e. there is no shell or similar to accept input from) then no prompts will be showed and the default action is to not overwrite any files. Use **NOPROMPT** to make UnTAR overwrite any existing files in this case.

This option is ignored if **LIST** is specified.

QUIET While extracting, do not show messages about the files that are being processed.

This option is ignored if **LIST** is specified.

CASE Makes the patterns (see below) case sensitive. This is needed because on Unix systems, file names are generally case sensitive, while on the Amiga they are not. You can then use this option to extract files with the same name but with different cases, should there be such files in the archive, as UnTAR does not process these in any special way (due to the fact that the OS does not care about the case of file names).

PATTERNS Any number of AmigaDOS patterns specifying which files in the archive to process. This applies both to list and extract operations. Note that the pattern matching is done on the names as shown by the LIST option, not the names of the created files. Most of the time these names are the same.

1.6 UnTAR.guide/Limitations

Limitations

Some of the limitations are due to lack of information on the tar archive format and possible variations. Current behaviour is based on the sources for untgz and GNU tar.

User and group ids for extracted files are ignored, as I'm not sure if they can be supported in any useful way on the Amiga. Let me know what you think.

Time zone handling isn't complete. If you have OS 2.1 or better, the builtin Locale system will be used to find the GMT offset, but this does not include daylight savings time. If you have OS 2.04, then time zones are ignored.

Links are ignored while extracting files. The main question is how they should be handled. What if the link target comes later in the archive? Should the links be soft or hard?

Some tar archives, and some files in some archives, will probably not work with UnTAR. E.g., GNU extensions are not supported.

File name handling might need improvements. Any leading "/"-char in the file name is removed, and colons (":") are replaced with underscores ("_") when files are extracted, but other chars might need to be taken care of.

1.7 UnTAR.guide/Acknowledgements

Acknowledgements

Pedro A. Aranda Gutiérrez Author of the original untgz program.

Jean-loup Gailly Co-author of **zlib** and made the unix adaption of untgz.

Mark Adler Co-author of zlib, used for decompression.

Martin Taillefer Author of AsyncIO.

1.8 UnTAR.guide/Glossary

Glossary

deflate Deflate is a common compression algorithm, e.g. used by zlib, gzip and pkzip. It is a variation of LZ77, where the compressed data is further compressed with huffman coding. Details can be found in rfc1951. The algorithm can compress data quite well and reasonably fast.

gzip Gzip, short for GNU Zip, is a program that compresses single files using the deflate algorithm.

ixemul ixemul is a shared Amiga library, written to make it easy to port Unix programs to the Amiga. One problem, at least for small programs, is that ixemul is quite large, around 150 kB.

tar Tar is an acronym for "tape archive", and was designed for storing backups on tapes. This means that the archive format as such isn't the most efficient one, to simplify handling with tapes. However, by using a program like gzip on a tar file, it becomes a compact way of storing several files.

Unix protection bits These bits consists of three groups of three bits each. Each group contains read, write and execute permissions for the file owner, users in the same group as the owner and other users, in that order.

zlib zlib is a link library to compress (and uncompress) data using the deflate algorithm.

1.9 UnTAR.guide/Source

Source

UnTAR.c is based on `untgz.c`, written by Pedro A. Aranda Gutiérrez and adapted to Unix by Jean-loup Gailly. However, fairly large changes have been made in order to make the Amiga version. Features have been added and error handling have been improved a lot. Some information regarding `tar` files come from studies of the GNU tar sources.

A part of `zlib` (`gzio.c`) have been modified as well, mainly to add AsyncIO support and remove code not needed by UnTAR. The result is the file `GZIO.c`. Zlib as such was compiled without any changes (note that `uInt` need to be set to a 32-bit int or long in `zconf.h`, or you might get problems when compiling it with SAS/C 6.58 for a 68020+ CPU).

Should anyone want to make a PPC version, please go ahead. I don't have a PPC card myself, so it is a bit difficult for me to do. But please contact me first, to avoid duplicated work.

1.10 UnTAR.guide/Author

Author

Magnus Holmgren Kvarnbergsvägen 5 SE-444 47 Stenungsund SWEDEN

E-mail: lear@algonet.se WWW: <http://www.algonet.se/~lear/>
