



## Appendix A2- Archives

Copyright ©2015 by Linda Marshall and Vreda Pieterse. All rights reserved.

### Contents

A2.1	Introduction . . . . .	2
A2.2	Creating tar archives . . . . .	2
A2.3	Extracting tar archives . . . . .	4
A2.4	Creating tarballs . . . . .	4
A2.5	Extracting tarballs . . . . .	5
	References . . . . .	5

## A2.1 Introduction

During 1969 development of an operating system called Unix was started by AT&T's Bell Labs. It was for this system that the tar (**t**ape **a**rchive) file format was created where it was used to write directly to sequential I/O devices for tape backup purposes [4]. The format was standardized by POSIX.1-1988 and later POSIX.1-2001. A file in tar format is not a compressed file. It is simply the concatenation of a number of files in a single file.

The GNU project was launched in 1984 to develop a complete Unix-like operating system which is free software. The name GNU was chosen, as a recursive acronym for “**G**NU's **N**ot **U**nix”. It provides the implementation of many tools that were originally only available on Unix systems. GNU tar is one of them [2].

In 1991, the first version of the Linux kernel was released by Linus Torvalds [1]. Early Linux kernel developers ported GNU code, including the GNU C Compiler and GNU tar to run on Linux. Later GNU developers adapted other parts of GNU to run on the Linux kernel producing the complete free OS we are currently using.

The GNU implementation of compression utility called gzip was created by Jean-Loup Gailly and Mark Adler. Version 0.1 was first publicly released on October 31, 1992 [3]. Version 1.0 followed in February 1993. The g in gzip stands for “gratis”, meaning *free*. The gzip utility compresses a single file. One important thing to remember about gzip is that, unlike tar, it replaces your original file with a compressed version.

By combining tar and gzip it is possible to create a compressed bundle. First tar is used to create a single file from the set of files and then gzip is applied to compress it. The final .tar.gz or .tgz file is usually called a tarball. Sometimes gzip truncates long filenames. However, it more often uses the .tgz extension instead of truncating a file with a .tar extension.

## A2.2 Creating tar archives

A single file containing any number of files can be created using the tar utility. It is possible to specify a number of files to be combined into a single .tar file by issuing a command like the following:

```
tar -c Account.cpp Account.h main.cpp -f Bank.tar
```

The above command concatenates the source files called Account.cpp, Account.h and main.cpp into a single target file called Bank.tar. The switches **-c** and **-f** respectively stands for **c**reate and **f**ile. The tar command must be issued using one of the eight defined operations of which **-c** is the most used of these operations. The **-f** switch must be followed by a space and the filename of the target file. The following two commands are valid variations of this command:

```
tar -f Bank.tar -c Account.cpp Account.h main.cpp
```

```
tar -cf Bank.tar Account.cpp Account.h main.cpp
```

Note that the flags may be presented in any order as long as their applicable parameters are specified properly. Note how the flags may be concatenated as shown in the last variation. It is important to note that these files must reside in the directory where the command is issued. If this is not the case the filenames may be specified with their complete paths. Likewise, the archive in the above command is created in the current directory, but may as well be created any valid location specified in the command.

It is also possible to add one or more new source files to an existing target archive by issuing a command like the following:

```
tar -r Customer.cpp Customer.h -f Bank.tar
```

The above command adds the source files called `Customer.cpp` and `Customer.h` to the existing target archive called `Bank.tar` which was created in the previous example. Here the first switch `r`, replaced the `c` switch as operation. `Připojit` is the Czech word that means append. It is another one of the eight defined tar operations.

A third tar operation is `update`. One can use it replace a file in an archive with a newer version of that file. This can be achieved by issuing a command like the following:

```
tar -u Customer.cpp -f Bank.tar
```

The above command will alter the existing target archive called `Bank.tar` to contain the newer version of the source file named `Customer.cpp`. If the original target did not contain a file called `Customer.cpp` it will simply be added.

Wild cards may also be used to specify the source files in any of the above commands. For example the following command will create a new target archive called `Project.tar` that contains all files with `.cpp` or `.h` extensions in the current directory:

```
tar -c *.cpp *.h -f Project.tar
```

When using wild cards in a tar command, it is advisable to add the `verbose` switch. This will cause the command to list the files that are processed. For example the following command will create the exact same archive as the above command. However, it will also list the files on the standard output.

```
tar -cv *.cpp *.h -f Project.tar
```

One can also precede every switch by `-` and switches may be listed in any order. For example the above command can also be written as:

```
tar -cf Project.tar -v *.cpp *.h
```

There are many more things one can specify when creating tar archives. For example one can preserve the hierarchical structure of the file collection that is archived. More detail regarding the usage of tar to create tar files can be found by typing `man tar` at the command prompt<sup>1</sup>.

---

<sup>1</sup>Quit the manual by pressing `q`

## A2.3 Extracting tar archives

One would expect that one needs an utility like **untar** to extract a tar archive. However, there is no such command. Instead one of the eight operations defined for the tar utility is to **extract** the target file. Therefore the following command will verbosely extract the entire archive called `Project.tar` in the current directory:

```
tar -xvf Project.tar
```

If the current directory already contains files with the same names than those in the archive, they are overwritten.

One can also selectively extract only specified files from an archive. For example the following command will only extract the header files in the given archive:

```
tar -xf archive.tar *.h
```

Here the verbose switch was omitted just to show that it is optional. This switch as well as any number of other switches can be added in the command. You are challenged to use `man tar` to find out how the following command will extract all the `.cpp` files in the given archive:

```
tar -xPC Project -vpf archive.tar *.cpp
```

You will notice that the switches for tar are case sensitive. Many switches also requires arguments. When a switch requires an argument, its argument follows the switch after a single space. Subsequent switches are then again preceded by `-`. Read more about them in the complete tar manual by typing `info tar` at the command prompt.

## A2.4 Creating tarballs

An archive that is written in compressed tar format (`.tar.gz` or `.tgz`) is called a tarball. Often code and documents are archived as a single tarball for easy distribution.

Firstly one combines the collection of files into a single uncompressed file using tar. Thereafter the resulting single file is compressed using gzip. This can be achieved by issuing commands like the following in succession:

```
tar -cvf Project.tar *.cpp *.h
gzip Project.tar
```

The first of the above two commands creates a tar archive consisting of all the `.cpp` and `.h` files in the current directory, while the second command uses gzip to compress this archive. The result will be a file called `Project.tar.gz` which is a compressed tar file containing the specified files.

The tar utility also provides a `gzip` switch to perform both these steps with a single command. The following command will create the same resulting `.tar.gz` file as would the above mentioned two commands:

```
tar -czvf Project.tar.gz *.cpp *.h
```

## A2.5 Extracting tarballs

The receiver of a tarball has to extract the archive before he/she can gain access to the files inside the tarball. Similar to when creating a tarball, the user can perform the operation in two steps by first unzipping the tar file and thereafter extracting it with commands like:

```
gunzip Project.tar.gz
tar -xvf Project.tar
```

The user can also choose to perform both these actions by issuing a single command by adding the `z` switch when extracting with the `tar` utility. For example:

```
tar -xzvf Project.tar.gz
```

## References

- [1] Michael Calore. 1991: Kid from helsinki foments linux revolution. <http://www.wired.com/thisdayintech/2009/08/0825-torvalds-starts-linux>, 2009. [Online; accessed 29-June-2010].
- [2] Richard Stallman. Linux and the gnu project. <http://www.gnu.org/gnu/linux-and-gnu.html>, 1997. [Online; accessed 29-June-2010].
- [3] Wikipedia. Gzip — wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Gzip&oldid=369888563>, 2010. [Online; accessed 29-June-2010].
- [4] Wikipedia. Tar (file format) — wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Tar\\_\(file\\_format\)&oldid=342169003](http://en.wikipedia.org/w/index.php?title=Tar_(file_format)&oldid=342169003), 2010. [Online; accessed 29-June-2010].