

15 Tar commands should try in Linux

On * nix operating systems, such as Linux, the commonly used utility is tar. The name of this command comes from ARchive Tape, because it was originally designed to back up data on tape.

Sometimes users need to pack multiple files and / or folders into a single file. It will be easier to transfer a file over the network and even all the files in the USB (but the more files, the slower the transfer will be). On operating systems like Windows, this is usually done by storing in .zip or .rar files. On * nix operating systems, such as Linux, the commonly used utility is tar. The name of this command comes from ARchive Tape, because it was originally designed to back up data on tape. Today, it can still be used for this purpose, but in most cases, it is simply the method of creating 'tarball'. The purpose of tarball is to facilitate uploading, downloading and moving data or keeping backups.

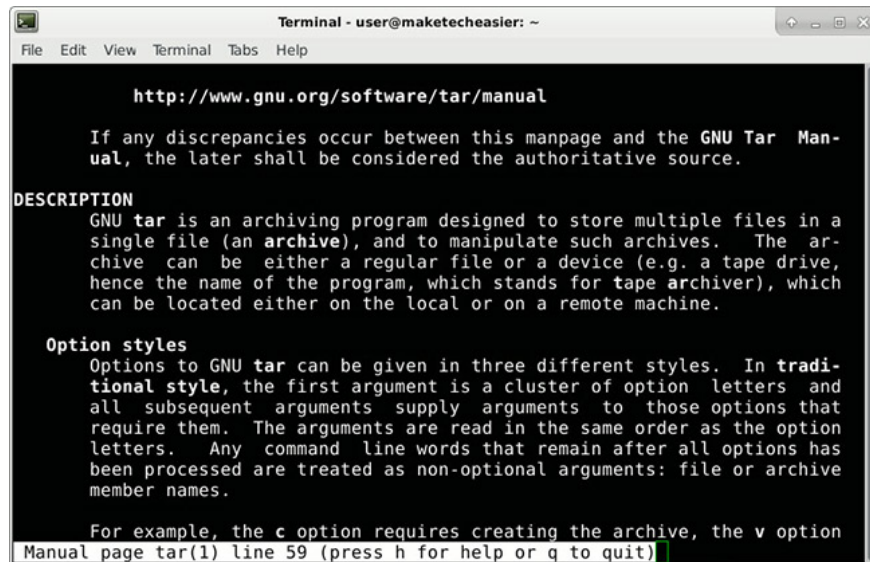
Tar commands are common in Linux

1. 1. Refer to the tar user manual
2. 2. Create a Tarball folder
3. 3. Add content to an existing Tarball
4. 4. List the contents of tar file
5. 5. Extract content from Tarball
6. 6. Create Tarball compression (gzip)
7. 7. Create smaller compressed Tarball (bzip)
8. 8. Extract a file or folder
9. 9. Extract only the files / folders that match a template
10. 10. Preserving access rights
11. 11. Delete files or folders from Tarball
12. 12. Exclude files or folders from being added to Tarball
13. 13. In a file system
14. 14. Preserve extended attributes
15. 15. Create new folders for the extracted files

1. Refer to the tar user manual

Each command in Linux has a tutorial page. You can refer to the tar user manual by entering the following command in the terminal:

```
man tar
```

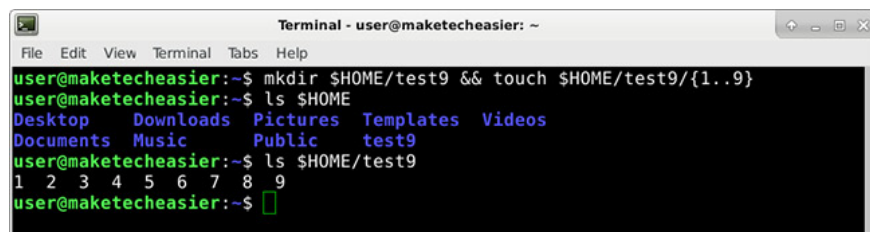


Look through the content using the arrow keys, PAGE UP and PAGE DOWN . Press **q** to exit the tutorial.

2. Create Tarball folder

Create a directory with 9 empty files to check tar commands:

```
mkdir $HOME/test9 && touch $HOME/test9/{1..9}
```



Now switch to the main directory:

```
cd
```

To transfer the contents of a directory to a tar file:

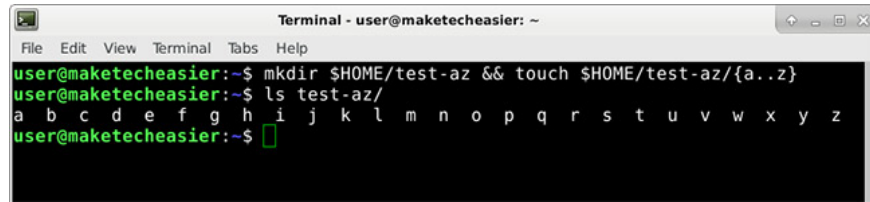
```
tar -cvf test.tar test9
```

1. **c** - stands for 'create'
2. **v** - **Verbose**, export the file currently added to the repository; Useful for time-consuming tasks and check progress
3. **f** - Inform the utility of creating a tar file, with the name specified as **test.tar**

3. Add content to an existing Tarball

Create another folder with empty files whose names are sorted from a to z:

```
mkdir $HOME/test-az && touch $HOME/test-az/{a..z}
```

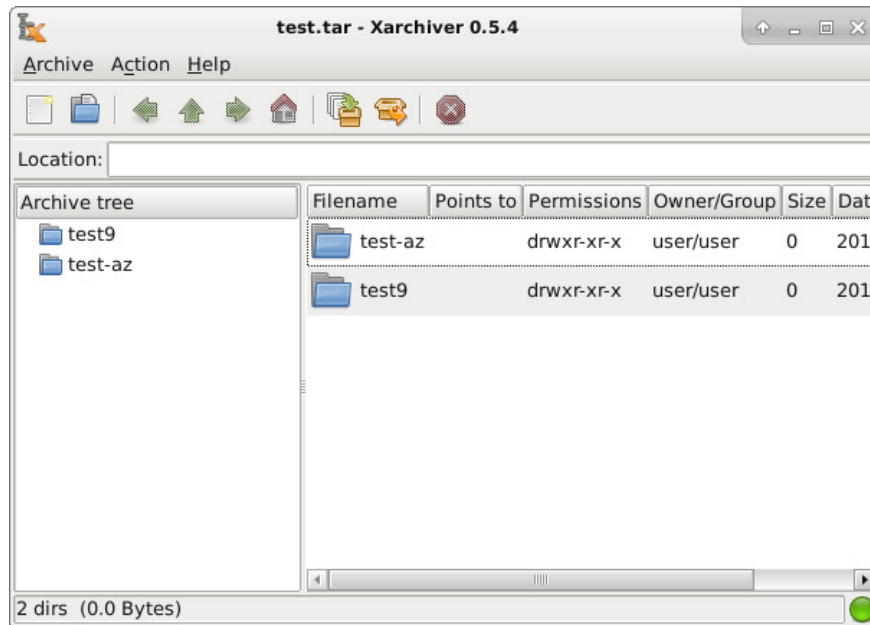


```
Terminal - user@maketecheasier: ~
File Edit View Terminal Tabs Help
user@maketecheasier:~$ mkdir $HOME/test-az && touch $HOME/test-az/{a..z}
user@maketecheasier:~$ ls test-az/
a b c d e f g h i j k l m n o p q r s t u v w x y z
user@maketecheasier:~$
```

Now, add it to the existing tar archive (parameter r):

```
tar -rvf test.tar test-az
```

If you open **test.tar** in the graphics storage manager, you will see the following content.



4. List the contents of tar file

Use the **-t** flag to list the contents of the tar file.

```
tar -tf test.tar
```

5. Extract content from Tarball

Content will be extracted in the current directory. Create a new folder and then move the content there:

```
mkdir $HOME/test-extract && cd $HOME/test-extract
```

Now extract the **test.tar** in the current directory:

```
tar -xvf $HOME/test.tar
```

```
Terminal - user@maketecheasier: ~/test-extract
File Edit View Terminal Tabs Help
test-az/r
test-az/w
test-az/l
test-az/h
test-az/t
test-az/o
test-az/k
test-az/y
test-az/b
test-az/g
test-az/i
test-az/e
test-az/d
test-az/u
test-az/v
user@maketecheasier:~/test-extract$ ls
test9 test-az
user@maketecheasier:~/test-extract$ ls *
test9:
1 2 3 4 5 6 7 8 9

test-az:
a b c d e f g h i j k l m n o p q r s t u v w x y z
user@maketecheasier:~/test-extract$
```

6. Create Tarball compression (gzip)

If you intend to transfer tar files over the network, making it smaller is a great idea to increase transfer speeds. Change back to the home directory.

```
cd
```

And create a compressed tar file:

```
tar -cvzf test.tar.gz test9
```

As can be seen, the only difference is that the **z** parameter is added to guide the tar to use gzip to compress the file.

```
Terminal - user@maketecheasier: ~
File Edit View Terminal Tabs Help
user@maketecheasier:~$ file test.tar
test.tar: POSIX tar archive (GNU)
user@maketecheasier:~$ file test.tar.gz
test.tar.gz: gzip compressed data, last modified: Thu Jan 17 17:10:09 2019, from Unix
user@maketecheasier:~$ ls -l
total 68
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Desktop
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Documents
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Downloads
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Music
drwxr-xr-x 2 user user 4096 Jan 17 12:06 Pictures
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Public
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Templates
drwxr-xr-x 2 user user 4096 Jan 17 11:42 test9
drwxr-xr-x 2 user user 4096 Jan 17 11:55 test-az
drwxr-xr-x 4 user user 4096 Jan 17 12:06 test-extract
-rw-r--r-- 1 user user 20480 Jan 17 11:59 test.tar
-rw-r--r-- 1 user user 213 Jan 17 12:10 test.tar.gz
drwxr-xr-x 2 user user 4096 Nov 25 23:59 Videos
user@maketecheasier:~$
```

7. Create smaller compressed Tarball (bzip)

Instead of the **z** parameter, users can use **j** to compress with bzip.

```
tar -cvjf test.tar.bz test9
```

This will continue to reduce the file size but take longer to compress and decompress.

8. Extract a file or folder

Just add the correct name of the file or directory, as it appears in the tar archive:

```
tar -xvf test.tar test9/7
```

This only extracts the **file test9/7** .

9. Extract only the files / folders that match a template

If you want to extract all the files ending with the **.jpg** extension :

```
tar -xvf test.tar --wildcards '*.jpg'
```

10. Preserving access rights

Access is not important when creating tarballs from documents, images and similar media files. But if you are backing up the system, you can add **p** parameter to maintain file access:

```
tar -cvpf test-allow.tar test9
```

11. Delete files or folders from Tarball

Only works on uncompressed tar files.

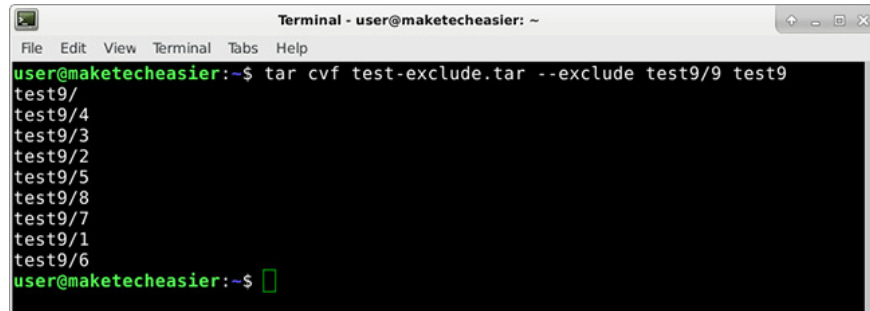
```
tar -f test.tar --delete test-az
```

This removes the **test-az** directory from the tarball, including the files contained therein:

tar f test.tar --delete test-az / z will only delete **test** file **test-az / z** .

12. Exclude files or folders from being added to Tarball

```
tar -cvf test-exclude.tar --exclude test9/9 test9
```

A terminal window titled "Terminal - user@maketecheasier: ~" showing the execution of a tar command. The command is `tar cvf test-exclude.tar --exclude test9/9 test9`. The output lists the files included in the archive: `test9/`, `test9/4`, `test9/3`, `test9/2`, `test9/5`, `test9/8`, `test9/7`, `test9/1`, and `test9/6`. The file `test9/9` is notably absent from the list. The prompt `user@maketecheasier:~$` is visible at the end of the output.

As you can see, file 9 has been removed from the tar archive. Can add many exclusion parameters:

```
tar cvf test-exclude.tar --exclude test9/9 --exclude test9/8 test9
```

13. In a file system

--one-file-system can be added in a command like:

```
tar -cvpzf backup.tar.gz --one-file-system /
```

This will back up the original file system, except for other file systems that you may have mounted in / **home**, / **mnt**, / **media**, / **proc**, etc.

14. Preserve extended attributes

Besides file access, some Linux distributions also have file extension attributes. To preserve all of them, add the following parameters: **--acls --selinux --xattrs** .

Example command:

```
tar -cvpzf backup.tar.gz --one-file-system --acls --selinux --xattrs /
```

15. Create new folders for the extracted files

Usually, tar will extract to the user's current directory. You can use **mkdir new_directory** and **cd new_directory** before using tar, if the directory you want to extract does not exist. But the tar guide does everything for you better:

```
tar -xvf test.tar --one-top-level=new_directory
```

These are the most commonly used tar commands. But this utility is very flexible and can be used in many different ways. If you don't like reading the instructions in the terminal, you can find the [HTML version](#) of the tar guide for easier reading.

See more:

1. Basic Shell commands in Linux
2. Kali Linux commands from AZ and commonly used commands
3. Combine all the most basic AutoCAD commands

You finished reading the article "**15 Tar commands should try in Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
