

How to Compress and Extract Files Using the tar Command on Linux

[Chris Hoffman](#) [@chrisbhoffman](#)

April 10, 2016, 6:40am EDT



The tar command on Linux is often used to create .tar.gz or .tgz archive files, also called “tarballs.” This command has a large number of options, but you just need to remember a few letters to quickly create archives with tar. The tar command can extract the resulting archives, too.

The GNU tar command included with Linux distributions has integrated compression. It can create a .tar archive and then compress it with gzip or bzip2 compression in a single command. That’s why the resulting file is a .tar.gz file or .tar.bz2 file.

Compress an Entire Directory or a Single File

Use the following command to compress an entire directory or a single file

on Linux. It'll also compress every other directory inside a directory you specify—in other words, it works recursively.

```
tar -czvf name-of-archive.tar.gz /path/to/directory-or-file
```

Here's what those switches actually mean:

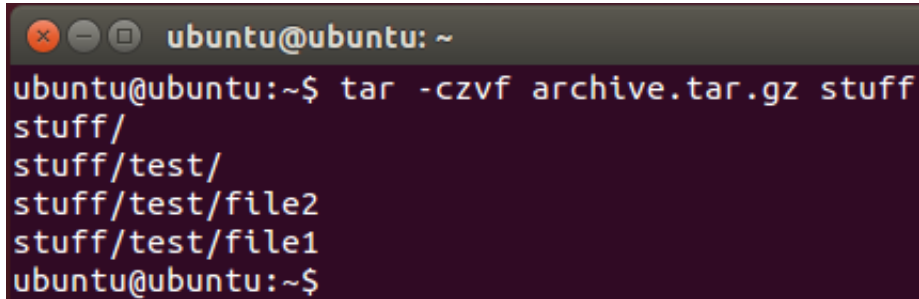
- **-c**: **C**reate an archive.
- **-z**: Compress the archive with **gzip**.
- **-v**: Display progress in the terminal while creating the archive, also known as "**v**erbose" mode. The **v** is always optional in these commands, but it's helpful.
- **-f**: Allows you to specify the **f**ilename of the archive.

Let's say you have a directory named "stuff" in the current directory and you want to save it to a file named archive.tar.gz. You'd run the following command:

```
tar -czvf archive.tar.gz stuff
```

Or, let's say there's a directory at /usr/local/something on the current system and you want to compress it to a file named archive.tar.gz. You'd run the following command:

```
tar -czvf archive.tar.gz /usr/local/something
```

A terminal window with a dark purple background and light green text. The window title is 'ubuntu@ubuntu: ~'. The command 'tar -czvf archive.tar.gz stuff' has been entered, and the output shows the contents of the 'stuff' directory being archived: 'stuff/', 'stuff/test/', 'stuff/test/file2', and 'stuff/test/file1'. The prompt 'ubuntu@ubuntu:~\$' is visible at the end of the output.

```
ubuntu@ubuntu:~$ tar -czvf archive.tar.gz stuff
stuff/
stuff/test/
stuff/test/file2
stuff/test/file1
ubuntu@ubuntu:~$
```

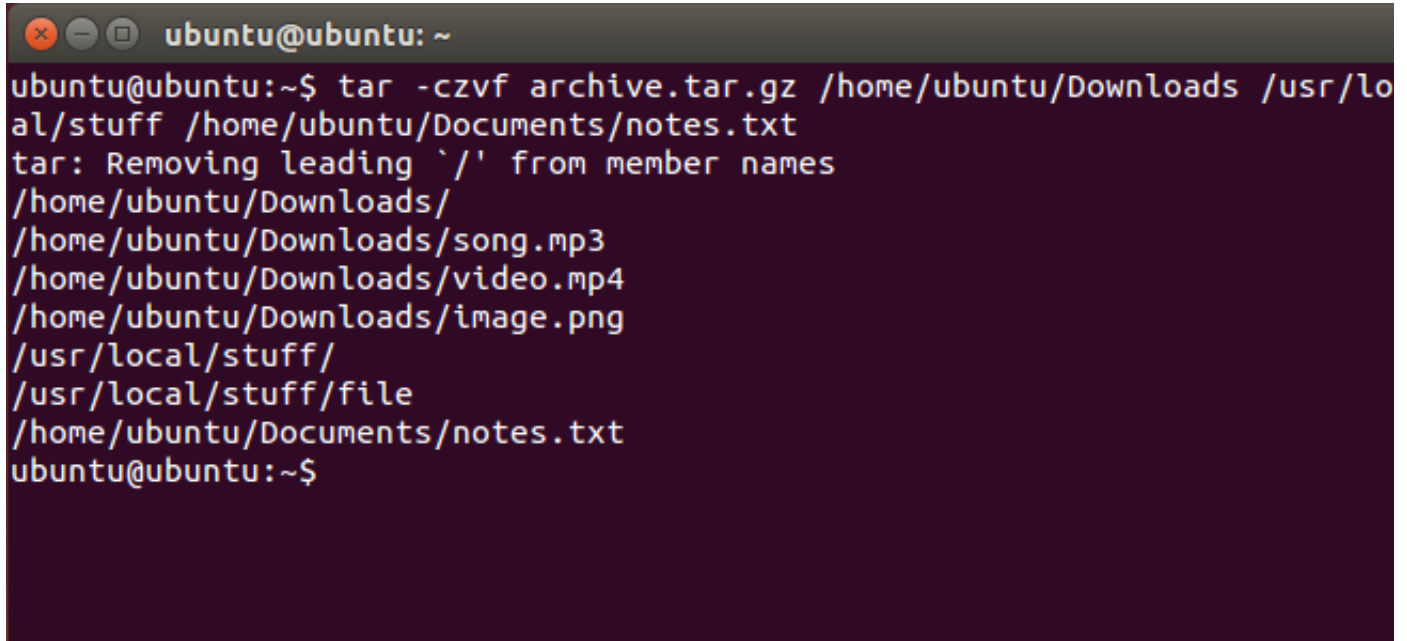
Compress Multiple Directories or Files at Once

RELATED: [How to Manage Files from the Linux Terminal: 11 Commands You Need to Know](#)

While tar is frequently used to compress a single directory, you could also use it to compress multiple directories, [multiple individual files](#), or both. Just provide a list of files or directories instead of a single one. For example, let's say you want to compress the /home/ubuntu/Downloads directory, the /usr/local/stuff directory, and the /home/ubuntu/Documents/notes.txt file. You'd just run the following command:

```
tar -czvf archive.tar.gz /home/ubuntu/Downloads /usr/local/stuff
```

Just list as many directories or files as you want to back up.

A terminal window with a dark purple background and light green text. The window title is 'ubuntu@ubuntu: ~'. The command 'tar -czvf archive.tar.gz /home/ubuntu/Downloads /usr/local/stuff /home/ubuntu/Documents/notes.txt' has been entered. The output shows the tar command removing the leading '/' from member names and listing the files being archived: /home/ubuntu/Downloads/, /home/ubuntu/Downloads/song.mp3, /home/ubuntu/Downloads/video.mp4, /home/ubuntu/Downloads/image.png, /usr/local/stuff/, /usr/local/stuff/file, and /home/ubuntu/Documents/notes.txt. The prompt returns to 'ubuntu@ubuntu:~\$'.

Exclude Directories and Files

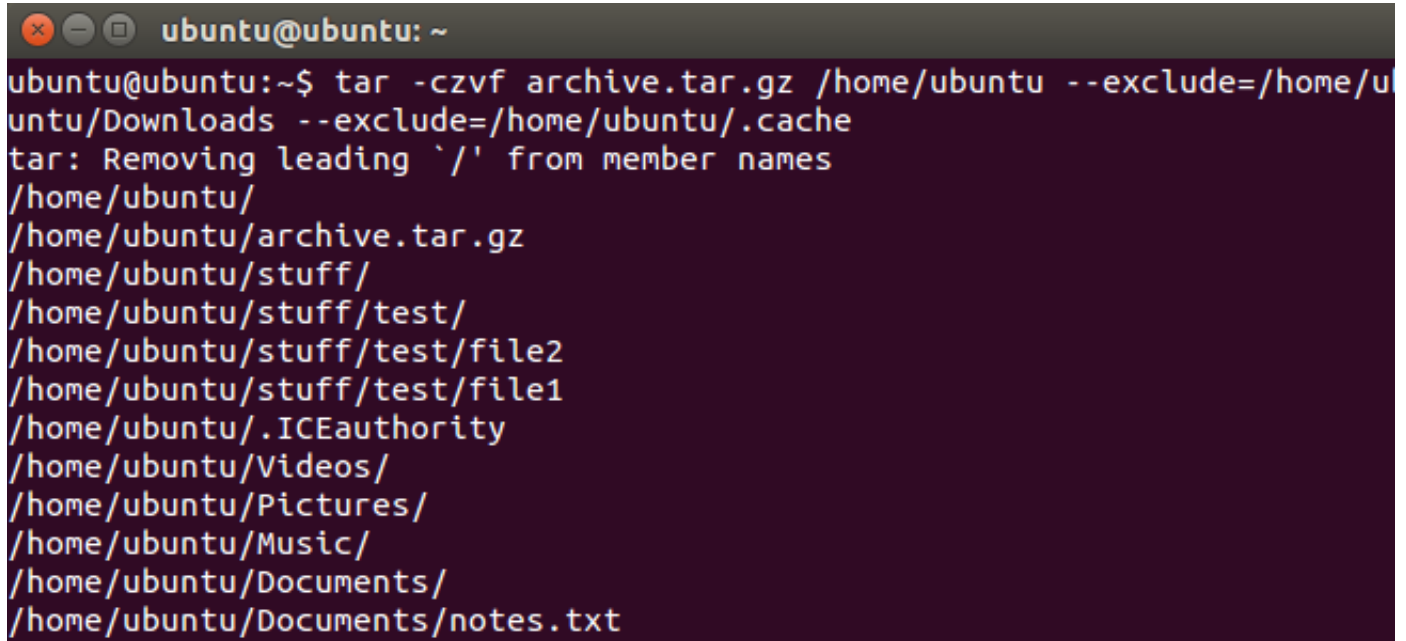
In some cases, you may wish to compress an entire directory, but not include certain files and directories. You can do so by appending an `--exclude` switch for each directory or file you want to exclude.

For example, let's say you want to compress `/home/ubuntu`, but you don't want to compress the `/home/ubuntu/Downloads` and `/home/ubuntu/.cache` directories. Here's how you'd do it:

```
tar -czvf archive.tar.gz /home/ubuntu --exclude=/home/ubuntu/Dow
```

The `--exclude` switch is very powerful. It doesn't take names of directories and files—it actually accepts patterns. There's a lot more you can do with it. For example, you could archive an entire directory and exclude all `.mp4` files with the following command:

```
tar -czvf archive.tar.gz /home/ubuntu --exclude=*.mp4
```

A terminal window with a dark purple background and light green text. The window title is 'ubuntu@ubuntu: ~'. The command entered is 'tar -czvf archive.tar.gz /home/ubuntu --exclude=/home/ubuntu/Downloads --exclude=/home/ubuntu/.cache'. The output shows the tar command removing the leading '/' from member names and listing the files and directories being archived: /home/ubuntu/, /home/ubuntu/archive.tar.gz, /home/ubuntu/stuff/, /home/ubuntu/stuff/test/, /home/ubuntu/stuff/test/file2, /home/ubuntu/stuff/test/file1, /home/ubuntu/.ICEauthority, /home/ubuntu/Videos/, /home/ubuntu/Pictures/, /home/ubuntu/Music/, /home/ubuntu/Documents/, and /home/ubuntu/Documents/notes.txt.

```
ubuntu@ubuntu:~$ tar -czvf archive.tar.gz /home/ubuntu --exclude=/home/ubuntu/Downloads --exclude=/home/ubuntu/.cache
tar: Removing leading '/' from member names
/home/ubuntu/
/home/ubuntu/archive.tar.gz
/home/ubuntu/stuff/
/home/ubuntu/stuff/test/
/home/ubuntu/stuff/test/file2
/home/ubuntu/stuff/test/file1
/home/ubuntu/.ICEauthority
/home/ubuntu/Videos/
/home/ubuntu/Pictures/
/home/ubuntu/Music/
/home/ubuntu/Documents/
/home/ubuntu/Documents/notes.txt
```

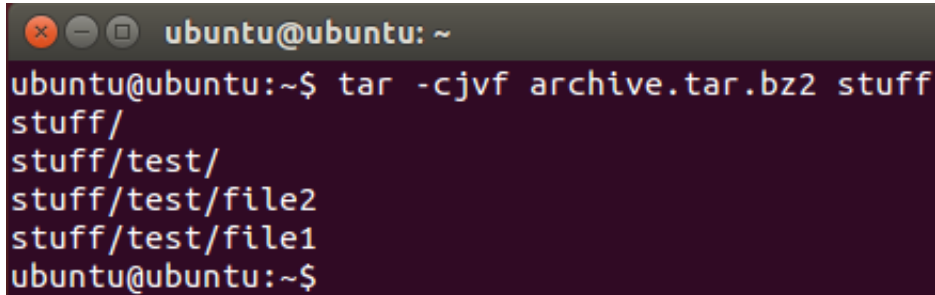
Use bzip2 Compression Instead

While gzip compression is most frequently used to create .tar.gz or .tgz files, tar also supports bzip2 compression. This allows you to create bzip2-compressed files, often named .tar.bz2, .tar.bz, or .tbz files. To do so, just replace the -z for gzip in the commands here with a -j for bzip2.

Gzip is faster, but it generally compresses a bit less, so you get a somewhat larger file. Bzip2 is slower, but it compresses a bit more, so you get a somewhat smaller file. Gzip is also more common, with some stripped-down Linux systems including gzip support by default, but not bzip2 support. In general, though, gzip and bzip2 are practically the same thing and both will work similarly.

For example, instead of the first example we provided for compressing the stuff directory, you'd run the following command:

```
tar -cjvf archive.tar.bz2 stuff
```

A terminal window with a dark purple background and light green text. The window title is 'ubuntu@ubuntu: ~'. The command 'tar -cjvf archive.tar.bz2 stuff' has been executed, and the output lists the contents of the archive: 'stuff/', 'stuff/test/', 'stuff/test/file2', and 'stuff/test/file1'. The prompt 'ubuntu@ubuntu:~\$' is visible at the end of the output.

```
ubuntu@ubuntu:~$ tar -cjvf archive.tar.bz2 stuff
stuff/
stuff/test/
stuff/test/file2
stuff/test/file1
ubuntu@ubuntu:~$
```

Extract an Archive

Once you have an archive, you can extract it with the tar command. The following command will extract the contents of archive.tar.gz to the current directory.

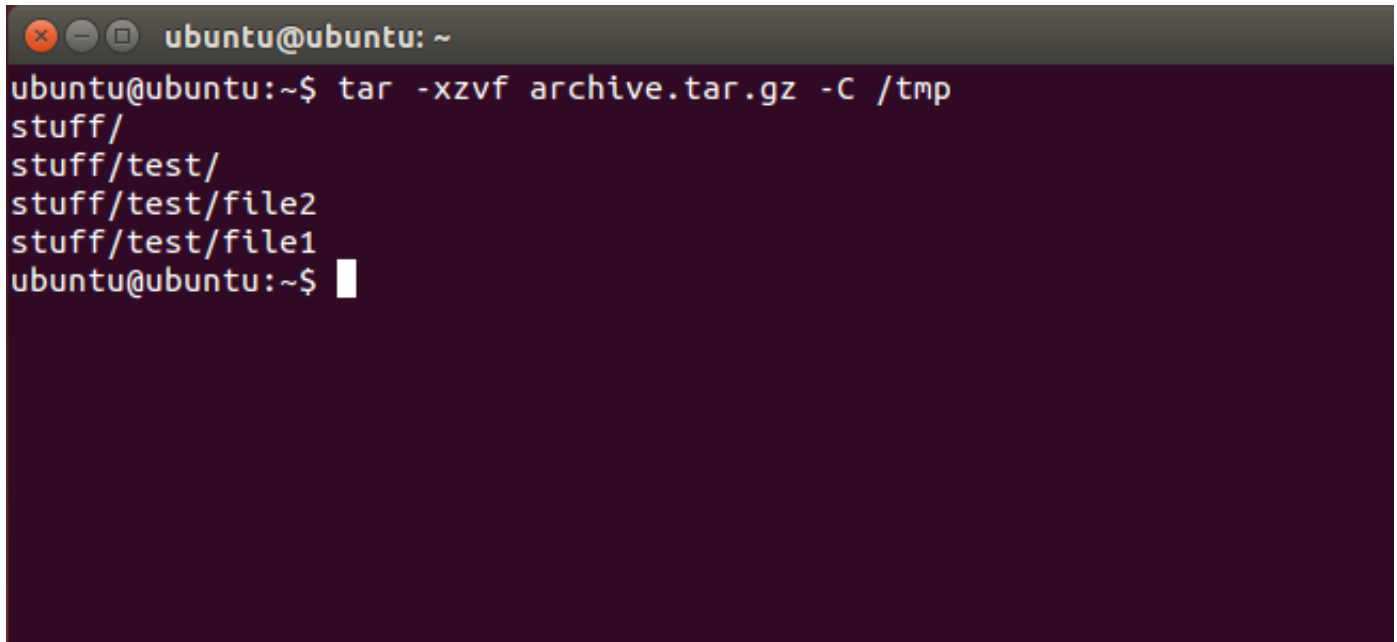
```
tar -xzvf archive.tar.gz
```

It's the same as the archive creation command we used above, except the `-x` switch replaces the `-c` switch. This specifies you want to **extract** an archive instead of create one.

You may want to extract the contents of the archive to a specific directory. You can do so by appending the `-C` switch to the end of the command. For example, the following command will extract the contents of the archive.tar.gz file to the /tmp directory.

```
tar -xzvf archive.tar.gz -C /tmp
```

If the file is a bzip2-compressed file, replace the “z” in the above commands with a “j”.

A terminal window with a dark purple background and light gray text. The window title is 'ubuntu@ubuntu: ~'. The command 'tar -xzvf archive.tar.gz -C /tmp' has been executed, and the output is displayed line by line: 'stuff/', 'stuff/test/', 'stuff/test/file2', and 'stuff/test/file1'. The prompt 'ubuntu@ubuntu:~\$' is visible at the bottom with a white cursor.

```
ubuntu@ubuntu:~$ tar -xzvf archive.tar.gz -C /tmp
stuff/
stuff/test/
stuff/test/file2
stuff/test/file1
ubuntu@ubuntu:~$
```

This is the simplest possible usage of the tar command. The command includes a large number of additional options, so we can’t possibly list them all here. For more information, run the **info tar** command at the shell to view the tar command’s detailed [information page](#). Press the **q** key to quit the information page when you’re done. You can also [read tar’s manual online](#).

If you’re using a graphical Linux desktop, you could also use the file-compression utility or file manager included with your desktop to create or extract .tar files. On Windows, you can extract and create .tar archives with the free [7-Zip](#) utility.

READ NEXT

- › [How to Customize Automatic Updates on iPhone and iPad](#)
- › [Can You Move Windows 10’s Notification Pop-ups?](#)
- › [CPUs Decoded: Understanding Intel’s Microarchitecture Names](#)

- › [How to Scan a File or Folder for Malware with Microsoft Defender on Windows 10](#)
- › [How to Use the whois Command on Linux](#)