

Tar Cheatsheet

By Dejan Panovski • Updated on Feb 13, 2026 • [Download PDF](#)

Quick reference for creating, listing, extracting, and compressing tar archives in Linux

Tar bundles files and directories into archive files, and it is commonly used with gzip, bzip2, or xz compression. This cheatsheet covers practical tar command patterns for backup, restore, and archive inspection.

Basic Syntax

Core tar command forms.

<code>tar -cf archive.tar files/</code>	Create archive
<code>tar -xf archive.tar</code>	Extract archive
<code>tar -tf archive.tar</code>	List archive contents
<code>tar -rf archive.tar file.txt</code>	Append file to archive (uncompressed only)
<code>tar --delete -f archive.tar file.txt</code>	Remove file from archive (uncompressed only)

Create Archives

Common archive creation patterns.

<code>tar -cf backup.tar /etc /var/www</code>	Archive multiple paths
<code>tar -cf project.tar project/</code>	Archive a directory
<code>tar -cvf project.tar project/</code>	Verbose create
<code>tar -cf archive.tar --files-from=filelist.txt</code>	Create from file list
<code>tar -cf archive.tar --exclude="*.log" app/</code>	Exclude pattern while creating

Extract Archives

Extract all files or selected paths.

<code>tar -xf archive.tar</code>	Extract in current directory
<code>tar -xvf archive.tar</code>	Verbose extraction
<code>tar -xf archive.tar -C /tmp/restore</code>	Extract to target directory
<code>tar -xf archive.tar path/in/archive.txt</code>	Extract one file
<code>tar -xf archive.tar "dir/*"</code>	Extract matching path pattern
<code>tar -xf archive.tar --strip-components=1</code>	Strip leading directory from paths

List and Inspect

Check what is inside an archive.

<code>tar -tf archive.tar</code>	List files
<code>tar -tvf archive.tar</code>	Detailed list with metadata
<code>tar -tf archive.tar head</code>	Preview first entries
<code>tar -tf archive.tar grep nginx</code>	Find matching archive paths
<code>tar --compare -f archive.tar</code>	Compare archive with filesystem

Compression Formats

Use gzip, bzip2, and xz with tar.

<code>tar -czf backup.tar.gz /data</code>	Create <code>.tar.gz</code>
<code>tar -xzf backup.tar.gz</code>	Extract <code>.tar.gz</code>
<code>tar -cjf backup.tar.bz2 /data</code>	Create <code>.tar.bz2</code>
<code>tar -xjf backup.tar.bz2</code>	Extract <code>.tar.bz2</code>
<code>tar -cJf backup.tar.xz /data</code>	Create <code>.tar.xz</code>
<code>tar -xJf backup.tar.xz</code>	Extract <code>.tar.xz</code>

Preserve Metadata

Keep permissions, ownership, and timestamps.

<code>tar -cpf system.tar /etc /var</code>	Archive system paths preserving permissions
<code>tar -xpf rootfs.tar</code>	Extract and preserve permissions
<code>tar --numeric-owner -cpf backup.tar /srv</code>	Store numeric UID/GID
<code>tar --same-owner -xpf backup.tar</code>	Keep original ownership on extract
<code>tar --atime-preserve=system -cf backup.tar /data</code>	Preserve file access times

Incremental Backups

Create snapshot-based incremental archives.

<code>tar --listed-incremental=snapshot.snar -cf full.tar /home</code>	Create initial full backup
<code>tar --listed-incremental=snapshot.snar -cf inc-1.tar /home</code>	Create next incremental backup
<code>tar -tf inc-1.tar</code>	List incremental archive entries
<code>tar -xf full.tar -C /restore</code>	Restore full backup
<code>tar -xf inc-1.tar -C /restore</code>	Apply incremental backup

Exclude Patterns

Skip files and directories during archive creation.

<code>tar -cf app.tar app/ --exclude="*.tmp"</code>	Exclude by extension
<code>tar -cf app.tar app/ --exclude="cache/*"</code>	Exclude path pattern
<code>tar -cf app.tar app/ --exclude=vcs</code>	Exclude VCS dirs (<code>.git</code> , etc.)
<code>tar -cf app.tar app/ --exclude-from=exclude.txt</code>	Exclude using a patterns file
<code>tar -czf logs.tar.gz /var/log --exclude="*.gz"</code>	Archive logs without already-compressed files

Safety and Verification

Run safer extraction and validation commands.

<code>tar -tvf archive.tar</code>	Inspect archive before extracting
<code>tar -xf archive.tar --keep-old-files</code>	Do not overwrite existing files
<code>tar -xf archive.tar --skip-old-files</code>	Skip files that already exist
<code>tar -xf archive.tar --warning=no-unknown-keyword</code>	Suppress noisy keyword warnings
<code>tar -tzf backup.tar.gz > /dev/null</code>	Quick integrity check for <code>.tar.gz</code>

Remote Transfer

Pipe tar over SSH for direct server-to-server transfers.

<code>tar -czf - /data ssh user@host "cat > backup.tar.gz"</code>	Archive and send to remote host
<code>ssh user@host "tar -czf - /data" tar -xzf - -C /local/path</code>	Pull and extract from remote host
<code>tar -cf - /data ssh user@host "tar -xf - -C /restore"</code>	Stream archive directly to remote

Common Options

Useful flags and what they do.

-c	Create archive
-x	Extract archive
-t	List archive contents
-f	Use archive file
-v	Verbose output
-z	Gzip compression
-j	Bzip2 compression
-J	XZ compression
-C	Change to directory before operation
-p	Preserve permissions

Related Guides

Use these articles for step-by-step tar workflows.

How to Create and Extract Archives Using the tar Command in Linux	Full tar tutorial with examples
How to Create tar.gz File	Focused <code>.tar.gz</code> creation guide
How to Extract/Unzip tar.gz File	<code>.tar.gz</code> extraction patterns