

mount Cheatsheet

By Dejan Panovski • Updated on Jul 24, 2026 •

Quick reference for the mount command: mount and unmount filesystems, mount options, UUID and label mounts, network shares, and /etc/fstab entries

The `mount` command attaches a filesystem to a directory in the tree, and `umount` detaches it. This cheatsheet covers basic syntax, listing mounts, common mount options, UUID and label lookups, network and special mounts, unmounting, and persistent `/etc/fstab` entries.

Basic Syntax

Core `mount` and `umount` command forms.

<code>mount DEVICE DIR</code>	Attach a device to a mount point
<code>mount -t TYPE DEVICE DIR</code>	Mount with an explicit filesystem type
<code>mount -o OPTIONS DEVICE DIR</code>	Mount with a comma-separated option list
<code>mount DIR</code>	Mount an entry already defined in <code>/etc/fstab</code>
<code>mount -a</code>	Mount <code>/etc/fstab</code> filesystems except <code>noauto</code> entries
<code>umount DIR</code>	Detach the filesystem mounted at <code>DIR</code>

Most mount operations require root privileges, so prefix them with `sudo`.

List Mounted Filesystems

Inspect what is currently mounted and where.

<code>mount</code>	List all mounts using the legacy display; prefer <code>findmnt</code>
<code>findmnt</code>	Show mounts as a readable tree
<code>findmnt /dev/sdb1</code>	Show where a specific device is mounted
<code>findmnt -t ext4,xfs</code>	List only the given filesystem types
<code>findmnt --fstab --verify</code>	Check <code>/etc/fstab</code> for errors before rebooting
<code>cat /proc/mounts</code>	Read active mounts in the current mount namespace
<code>lsblk -f</code>	List block devices with type, label, and UUID
<code>df -hT</code>	Show mounted filesystems with type and free space

Common Mount Options

Values passed to `-o` or listed in the fourth `/etc/fstab` field.

<code>defaults</code>	Common defaults are <code>rw,suid,dev,exec,auto,nouser,async</code> ; exact defaults vary
<code>ro / rw</code>	Mount read-only or read-write
<code>noexec</code>	Block execution of binaries on the filesystem
<code>nosuid</code>	Ignore setuid and setgid bits
<code>nodev</code>	Ignore device files on the filesystem
<code>noatime</code>	Skip access-time updates, which reduces disk writes
<code>sync / async</code>	Write synchronously or let the kernel buffer writes
<code>noauto</code>	Skip this entry during <code>mount -a</code> and at boot
<code>user</code>	Allow a non-root user to mount; implies <code>noexec,nosuid,nodev</code> unless overridden
<code>nofail</code>	Continue booting when the device is missing
<code>uid=1000,gid=1000</code>	Set owner and group on filesystems without Unix permissions
<code>remount</code>	Change options on an already mounted filesystem

Example: `sudo mount -o remount,rw /` remounts the root filesystem read-write.

Mount by UUID or Label

Device names such as `/dev/sdb1` can change between boots, so identify the filesystem instead.

<code>lsblk -f</code>	Show the UUID and label of every filesystem
<code>sudo blkid</code>	Print device type, label, and UUID
<code>sudo blkid /dev/sdb1</code>	Print the attributes of one device
<code>sudo mount UUID=1a2b3c4d-... /mnt/data</code>	Mount by filesystem UUID
<code>sudo mount LABEL=backup /mnt/backup</code>	Mount by filesystem label
<code>sudo mount -U 1a2b3c4d-... /mnt/data</code>	Short form of the UUID lookup
<code>sudo mount -L backup /mnt/backup</code>	Short form of the label lookup
<code>sudo e2label /dev/sdb1 backup</code>	Set a label on an ext2, ext3, or ext4 filesystem

Prefer `UUID=` or `LABEL=` for persistent local mounts in `/etc/fstab` because device names can change.

Filesystem Types

Values for `-t` and for the third `/etc/fstab` field.

<code>ext4</code>	Widely used general-purpose Linux filesystem
<code>xfs</code>	High-performance journaling filesystem, default on RHEL
<code>btrfs</code>	Copy-on-write filesystem with snapshots and subvolumes
<code>vfat</code>	FAT32, used by USB drives and EFI system partitions
<code>exfat</code>	Large-file FAT variant for USB drives and SD cards
<code>ntfs3</code>	In-kernel NTFS driver available on supported kernels
<code>iso9660</code>	Optical disc and ISO image filesystem
<code>tmpfs</code>	In-memory filesystem backed by RAM and swap
<code>nfs</code>	Network File System share
<code>cifs</code>	SMB or Windows network share
<code>auto</code>	Let <code>mount</code> detect the type

Network and Special Mounts

Mount forms beyond a plain local partition.

<code>sudo mount -t nfs server:/srv/data /mnt/data</code>	Mount an NFS export
<code>sudo mount -t cifs //server/share /mnt/share -o username=user</code>	Mount a Windows or Samba share
<code>sudo mount -t cifs //server/share /mnt/share -o credentials=/etc/smb-credentials</code>	Mount CIFS with a credentials file
<code>sshfs user@host:/remote/dir /mnt/remote</code>	Mount a remote directory over SSH
<code>sudo mount -o loop image.iso /mnt/iso</code>	Mount an ISO image through a loop device
<code>sudo mount --bind /src /dst</code>	Expose an existing directory at a second path
<code>sudo mount --rbind /src /dst</code>	Bind a directory together with its submounts
<code>sudo mount -t tmpfs -o size=512M tmpfs /mnt/ram</code>	Create a 512 MB RAM-backed filesystem

Keep CIFS credentials outside project directories and version control. Protect the file with `sudo chmod 600 /etc/smb-credentials`.

Unmounting

Detach a filesystem and deal with a busy mount point.

<code>sudo umount /mnt/data</code>	Unmount by mount point
<code>sudo umount /dev/sdb1</code>	Unmount by device (obsolete; use the mount point)
<code>sudo umount -R /mnt/data</code>	Unmount a mount point and everything below it
<code>sudo umount -l /mnt/data</code>	Lazy unmount: detach now, release when no longer in use
<code>sudo umount -f /mnt/share</code>	Force an unmount, mainly for unreachable network shares
<code>sudo umount -a -t nfs</code>	Unmount every NFS filesystem
<code>lsuf +f -- /mnt/data</code>	List open files under the mount point
<code>fuser -mv /mnt/data</code>	Show processes and users holding the mount point
<code>sudo fuser -km /mnt/data</code>	Kill every process using the mount point; destructive

Review `fuser -mv` output before using `fuser -km`. Use lazy or forced unmounts only when a normal unmount cannot work: `-f` can lose unwritten data, while `-l` hides the path before all references are released.

Persistent Mounts in /etc/fstab

Each line has six space-separated fields.

1. Device	UUID=, LABEL=, a device path, or a remote share
2. Mount point	Existing directory, or none for swap
3. Type	Filesystem type, such as ext4 , nfs , or cifs
4. Options	Comma-separated mount options, or defaults
5. Dump	0 in nearly all cases; 1 marks the filesystem for dump
6. Pass	Order of fsck at boot: 1 for root, 2 for others, 0 to skip

Example: `UUID=1a2b3c4d-... /mnt/data ext4 defaults,nofail 0 2`

<code>sudo findmnt --fstab --verify</code>	Validate the file before rebooting
<code>sudo mount -a</code>	Apply mountable entries after validating <code>/etc/fstab</code>
<code>sudo systemctl daemon-reload</code>	Regenerate the systemd mount units after an edit
<code>sudo mount /mnt/data</code>	Mount one entry using only its mount point

Use `nofail` only for optional mounts. Add `_netdev` when a network-backed filesystem is not recognized as network-dependent from its type.

Troubleshooting

Common mount errors and what to check.

target is busy	Find the holder with <code>fuser -mv DIR</code> or <code>lsof +f -- DIR</code> , then close it
unknown filesystem type	Install the required helper or driver, such as <code>ntfs-3g</code> , <code>nfs-common</code> or <code>nfs-utils</code> , or <code>cifs-utils</code> ; confirm kernel support for exFAT
wrong fs type, bad option, bad superblock	Confirm the type with <code>lsblk -f</code> and check <code>dmesg</code> . If repair is needed, unmount it and use the correct filesystem checker
mount point does not exist	Create the directory first with <code>sudo mkdir -p DIR</code>
permission denied on CIFS	Verify credentials and server share permissions; <code>uid=</code> and <code>gid=</code> only control local ownership
special device does not exist	Recheck the UUID with <code>blkid</code> ; the device name may have changed
Files are hidden after mounting	The mounted filesystem covers existing directory contents; unmount it and inspect the underlying path
Boot drops to emergency mode	Remount root with <code>mount -o remount,rw /</code> , fix the bad <code>/etc/fstab</code> line, and validate it; add <code>nofail</code> only if the mount is optional

Related Guides

Use these guides and cheatsheets for complete storage workflows.

How to Mount and Unmount File Systems in Linux	Full guide to <code>mount</code> and <code>umount</code> with practical examples
Understanding the /etc/fstab File in Linux	Field-by-field breakdown of persistent mount entries
How to Mount an NFS Share in Linux	Install the client, mount a share, and troubleshoot errors
How to Mount a Windows Share on Linux Using CIFS	Credentials files, ownership, and permissions for SMB shares
How to Mount ISO File on Linux	Loop devices and the graphical mounting method
How to Use SSHFS to Mount Remote Directories over SSH	Mount a remote directory and browse it like a local one
df Cheatsheet	Check free space on mounted filesystems
fdisk Cheatsheet	Partition a disk before creating a filesystem