systemd-mount

- Description
- Options
- Exit status
- The udev Database
- Example
- See Also

systemd

systemd-mount

1

systemd-mount

systemd-umount

Establish and destroy transient mount or auto-mount points

systemd-mount

OPTIONS

WHAT

WHERE

systemd-mount

OPTIONS

--tmpfs

NAME

WHERE

systemd-mount

OPTIONS

--list

systemd-mount

OPTIONS

--umount

WHAT|WHERE

Description

`systemd-mount` may be used to create and start a transient `.mount` or `.automount` unit of the file system <WHAT> on the mount point <WHERE>.

In many ways, `systemd-mount` is similar to the lower-level `mount(8)` command, however instead of executing the mount operation directly and immediately, `systemd-mount` schedules it through the service manager job queue, so that it may pull in further dependencies (such as parent mounts, or a file system checker to execute a priori), and may make use of the auto-mounting logic.

The command takes either one or two arguments. If only one argument is specified it should refer to a block device or regular file containing a file system (e.g. `/dev/sdb1` or `/path/to/disk.img`). The block device or image file is then probed for a file system label and other metadata, and is mounted to a directory below `/run/media/system/` whose name is generated from the file system label. In this mode the block device or image file must exist at the time of invocation of the command, so that it may be probed. If the device is found to be a removable block device (e.g. a USB stick), an automount point is created instead of a regular mount point (i.e. the `--automount=` option is implied, see below). If the option `--tmpfs` is specified, then the argument is interpreted as the path where the new temporary file system shall be mounted.

If two arguments are specified, the first indicates the mount source (the <WHAT>) and the second indicates the path to mount it on (the <WHERE>). In this mode no probing of the source is attempted, and a backing device node does not have to exist. However, if this mode is combined with `--discover`, device node probing for additional metadata is enabled, and – much like in the single-argument case discussed above – the specified device has to exist at the time of invocation of the command.

Use the `--list` command to show a terse table of all local, known block devices with file systems that may be mounted with this command.

`systemd-umount` can be used to unmount a mount or automount point. It is the same as `systemd-mount --umount`.

Options

The following options are understood:

`--no-block`
Do not synchronously wait for the requested operation to finish. If this is not specified, the job will be verified, enqueued and `systemd-mount` will wait until the mount or automount unit's start-up is completed. By passing this argument, it is only verified and enqueued.

`-l`; `--full`
Do not ellipsize the output when `--list` is specified.

`--quiet`; `-q`
Suppresses additional informational output while running.

`--discover`
Enable probing of the mount source. This switch is implied if a single argument is specified on the command line. If passed, additional metadata is read from the device to enhance the unit to create. For example, a descriptive string for the transient units is generated from the file system label and device model. Moreover, if a removable block device (e.g. USB stick) is detected an automount unit instead of a regular mount unit is created, with a short idle timeout, in order to ensure the file-system is placed in a clean state quickly after each access.

`--type=`; `-t`
Specifies the file system type to mount (e.g. `vfat` or `ext4`). If omitted or set to `auto`, the file system type is determined automatically.

`--options=`; `-o`
Additional mount options for the mount point.

`--owner=USER`
Let the specified user <USER> own the mounted file system. This is done by appending `uid=` and `gid=` options to the list of mount options. Only certain file systems support this option.

`--fsck=`
Takes a boolean argument, defaults to on. Controls whether to run a file system check immediately before the mount operation. In the automount case (see `--automount=` below) the check will be run the moment the first access to the device is made, which might slightly delay the access.

`--description=`
Provide a description for the mount or automount unit. See `Description=` in `systemd.unit(5)`.

`--property=`; `-p`
Sets a unit property for the mount unit that is created. This takes an assignment in the same format as `systemctl(1)`'s `set-property` command.

`--automount=`
Takes a boolean argument. Controls whether to create an automount point or a regular mount point. If true an automount point is created that is backed by the actual file system at the time of first access. If false a plain mount point is created that is backed by the actual file system immediately. Automount points have the benefit that the file system stays unmounted and hence in clean state until it is first accessed. In automount mode the `--timeout-idle-sec=` switch (see below) may be used to ensure the mount point is unmounted automatically after the last access and an idle period passed.

If this switch is not specified, it defaults to false. If not specified and `--discover` is used (or only a single argument passed, which implies `--discover`, see above), and the file system block device is detected to be removable, it is set to true, in order to increase the chance that the file system is in a fully clean state if the device is unplugged abruptly.

`-A`
Equivalent to `--automount=yes`.

`--timeout-idle-sec=`
Takes a time value that controls the idle timeout in automount mode. If set to `infinity` (the default) no automatic unmounts are done. Otherwise, the file system backing the automount point is detached after the last access and the idle timeout passed. See `systemd.time(7)` for details on the time syntax supported. This option has no effect if only a regular mount is established, and automounting is not used.

Note that if `--discover` is used (or only a single argument passed, which implies `--discover`, see above), and the file system block device is detected to be removable, `--timeout-idle-sec=1s` is implied.

`--automount-property=`
Similar to `--property=`, but applies additional properties to the automount unit created, instead of the mount unit.

`--bind-device`
Controls whether the generated mount/automount unit shall be bound to the backing device's lifetime. This setting is especially useful in automount mode. If set, the units will be stopped automatically when the backing device vanishes. By default, the automount unit stays around, and subsequent accesses will block until backing device is replugged. This option has no effect in case of non-device mounts, such as network or virtual file system mounts.

If `--discover` is used (or only a single argument passed, which implies `--discover`, see above), and the file system block device is detected to be removable, this option is implied.

Note that mount units by default gain a `Requires=` dependency on the backing device. This behavior can be controlled via `x-systemd.device-bound=` mount option, see `systemd.mount(5)` for details. In particular, `x-systemd.device-bound=no` takes precedence over this option, which suppresses device dependencies both in the generated mount units and what's implied by service manager.

`--list`
Instead of establishing a mount or automount point, print a terse list of block devices containing file systems that may be mounted with `systemd-mount`, along with useful metadata such as labels, etc.

`-u`; `--umount`
Stop the mount and automount units corresponding to the specified mount points <WHERE> or the devices <WHAT>. `systemd-mount` with this option or `systemd-umount` can take multiple arguments which can be mount points, devices, `/etc/fstab` style node names, or backing files corresponding to loop devices, like `systemd-mount --umount /path/to/umount /dev/sda1 UUID=xxxxxx-xxxx LABEL=xxxxx /path/to/disk.img`. Note that when `-H` or `-M` is specified, only absolute paths to mount points are supported.

`-G`; `--collect`
Unload the transient unit after it completed, even if it failed. Normally, without this option, all mount units that mount and failed are kept in memory until the user explicitly resets their failure state with `systemctl reset-failed` or an equivalent command. On the other hand, units that stopped successfully are unloaded immediately. If this option is turned on the "garbage collection" of units is more aggressive, and unloads units regardless of whether they exited successfully or failed. This option is a shortcut for `--property=CollectMode=inactive-or-failed`, see the explanation for `CollectMode=` in `systemd.unit(5)` for further information.

`-T`; `--tmpfs`
Create and mount a new `tmpfs` file system on <WHERE>, with an optional <NAME> that defaults to `tmpfs`.

The file system is mounted with the top-level directory mode determined by the `umask(2)` setting of the caller, i.e. `rwxrwxrwx` masked by the umask of the caller. This matches what `mkdir(1)` does, but is different from the kernel default of `rwxrwxrwxt`, i.e. a world-writable directory with the sticky bit set.

`--canonicalize=`
Controls whether the specified path shall be canonicalized on the client side before requesting the operation or not. Takes a boolean parameter, defaults to true. Note that for non-local operation (i.e. when `--machine=` or --`--host=` are used) canonicalization is implicitly turned off.

Canonicalization of path entails resolving of symlinks, `..` path elements and `LABEL=`/`UUID=` style device node expansion. If canonicalization is disabled and the path contains a symlink element, `..` , or a `LABEL=`/`UUID=`/… expansion the operation will fail.

Exit status

On success, 0 is returned, a non-zero failure code otherwise.

The udev Database

If `--discover` is used, `systemd-mount` honors a couple of additional udev properties of block devices:

`SYSTEMD_MOUNT_OPTIONS=`
The mount options to use, if `--options=` is not used.

`SYSTEMD_MOUNT_WHERE=`
The file system path to place the mount point at, instead of the automatically generated one.

Example

Use a udev rule like the following to automatically mount all USB storage plugged in:

```
ACTION=="add", SUBSYSTEMS=="usb", SUBSYSTEM=="block", ENV{ID_FS_USAGE}=="filesystem", \
        RUN{program}+="/usr/bin/systemd-mount --no-block --automount=yes --collect $devnode"
```

See Also

`systemd(1)`, `mount(8)`, `systemctl(1)`, `systemd.unit(5)`, `systemd.mount(5)`, `systemd.automount(5)`, `systemd-run(1)`

□□□ #1
□ □□□ □□□ 2026-10-31 15:00:46 UTC
□ □□□ □□□ 2026-10-31 15:00:46 UTC