



[Translation\(s\)](#): [English](#) - [Français](#) - [Italiano](#) - [Русский](#) - [Español](#)

The **fstab** (`/etc/fstab`) (or file systems table) file is a system configuration file on Debian systems. The `fstab` file typically lists all available disks and disk partitions, and indicates how they are to be initialized or otherwise integrated into the overall system's file system.

## Contents

1. [Example](#)

2. [Field definitions](#)

3. [Defining filesystems](#)

1. [Kernel naming](#)

2. [UUIDs](#)

3. [Labels](#)

4. [See also](#)

## Example

```
# <file system>      <dir>      <type>      <options>
/dev/sda1            /           ext4         defaults
/dev/hdxx            /usr        ext4         defaults
/dev/sda5            swap        swap         defaults
```

It is not necessary to list `/proc` and `/sys` in the `fstab` unless some special options are needed. The boot system will always mount them.

## Field definitions

`/etc/fstab` contains the following fields separated by a space or tab:

```
<file system>  <dir>  <type>  <options>      <dump>  <pass
```

- **<file systems>** - defines the storage device (i.e. `/dev/sda1`).
- **<dir>** - tells the mount command where it should mount the `<file system>` to.

- **<type>** - defines the file system type of the device or partition to be mounted. Many different file systems are supported. Some examples are: ext2, ext3, reiserfs, xfs, jfs, smbfs, iso9660, vfat, ntfs, swap, and auto. The 'auto' type lets the mount command to attempt to guess what type of file system is used, this is useful for removable devices such as CDs and DVDs.
- **<options>** - define particular options for filesystems. Some options relate only to the filesystem itself. Some of the more common options are:
  - **auto** - file system will mount automatically at boot, or when the command 'mount -a' is issued.
  - **noauto** - the filesystem is mounted only when you tell it to.
  - **exec** - allow the execution binaries that are on that partition (default).
  - **noexec** - do not allow binaries to be executed on the filesystem.
  - **ro** - mount the filesystem read only.
  - **rw** - mount the filesystem read-write.
  - **sync** - I/O should be done synchronously.
  - **async** - I/O should be done asynchronously.
  - **flush** - specific option for FAT to flush data more often, thus making copy dialogs or progress bars to stays up until things are on the disk.
  - **user** - permit any user to mount the filesystem (implies noexec,nosuid,nodev unless overridden).
  - **nouser** - only allow root to mount the filesystem (default).
  - **defaults** - default mount settings (equivalent to rw,suid,dev,exec,auto,nouser,async).
  - **suid** - allow the operation of suid, and sgid bits. They are mostly used to allow users on a computer system to execute binary executables with temporarily elevated privileges in order to perform a specific task.
  - **nosuid** - block the operation of suid, and sgid bits.
  - **noatime** - do not update inode access times on the filesystem. Can help performance.
  - **nodiratime** - do not update directory inode access times on the filesystem. Can help performance. You do not need to enable this flag if you have already enabled **noatime**.
  - **relatime** - update inode access times relative to modify or change time. Access time is only updated if the previous access time was earlier than the current modify or change time (similar to noatime, but doesn't break mutt or other applications that need to know if a file has been read since the last time it was modified). Can help performance.
- **<dump>** - is used by the dump utility to decide when to make a backup. When installed, dump checks the entry and uses the number to decide if

a file system should be backed up. Possible entries are 0 and 1. If 0, dump will ignore the file system, if 1, dump will make a backup. Most users will not have dump installed, so they should put 0 for the `<dump>` entry.

- **<pass>** fsck reads the `<pass>` number and determines in which order the file systems should be checked. Possible entries are 0, 1, and 2. The root file system should have the highest priority, 1, all other file systems you want to have checked should get a 2. File systems with a `<pass>` value 0 will not be checked by the fsck utility.

## Defining filesystems

You can define the filesystems in the `/etc/fstab` configuration in three different ways: by kernel naming descriptors, by UUID, or by labels. The advantage of using UUIDs or labels is that they are not dependent on disk order. This is useful if you change your storage device order in the BIOS, you switch storage device cabling, or because some BIOS's may occasionally change the order of storage devices.

LVM snapshots can result in duplicate UUIDs and labels, so LVM logical volumes should always be identified by logical volume name (`/dev/mapper/name`).

### Kernel naming

You can get kernel naming descriptors using fdisk:

```
# fdisk -l
...
    Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         19458    156290016+  83  Linux
```

An example `/etc/fstab` using the kernel naming:

```
# <file system>      <dir>          <type>        <options>
/dev/sda1             /               ext4           defaults
```

### UUIDs

UUIDs are generated by the make-filesystem utilities (mkfs.\*) when you create a filesystem. blkid will show you the UUIDs of mounted devices and partitions:

```
# blkid
/dev/sda1: UUID="6a60524d-061d-454a-bfd1-38989910eccd" TYPE="
```

An example `/etc/fstab` using the UUID identifiers:

```
# <file system>      <dir>          <type>      <options>
UUID=6a60524d-061d-454a-bfd1-38989910eccd      /
```

## Labels

The device or partition is required to be labelled first. To do this, you can use common applications like `gparted` to label partitions or you can use `e2label` to label `ext2`, `ext3`, and `ext4` partitions. Keep in mind that not all file systems have labelling support (e.g. FAT file systems). Labels can be up to 16 characters long. Labels should be unambiguous, meaning that each label should be original to prevent any possible conflicts.

A device or partition must not be mounted before attempting to label them. Initially you will need to boot from a Live(CD/DVD/USB) before you can label with a `gparted`-like application or for `ext` partitions you can use `e2label`:

```
# e2label /dev/<disk-or-partition> Debian
```

An example `/etc/fstab` using the labels:

```
# <file system>      <dir>          <type>      <options>
LABEL=Debian        /              ext4        defaults
```

## See also

- [Hardware](#)

- [Part-UUID](#) (should be merging here)

---

[CategorySystemAdministration](#) | [CategoryStorage](#)