



Understanding fstab

/etc/fstab is a system configuration file and is used to tell the Linux kernel which partitions (file systems) to mount and where on the file system tree.

[bodhi.zazen](http://bodhi.zazen.com)

Ubuntu Guru



ubuntu forums admin





Table of Contents

[Introduction / Mount](#).....3

["fstab Syntax" - Syntax and fstab options](#).....4

[How to label, FAT and Linux file systems](#).....10

[Examples of, FAT and Linux native files systems](#).....15

[References](#).....17

Sorry this is such a long post.

I added much of this information to the Ubuntu wiki.

[Ubuntu Wiki : fstab](#)

[bodhi.zazen](#)

Ubuntu Guru

 ubuntu forums admin



Introduction

/etc/fstab is a system configuration file and is used to tell the Linux kernel which partitions (file systems) to mount and where on the file system tree.

/etc/mtab is an index of all mounted partitions/file systems.

Note: See [references section](#) at the end of this how to for useful links.

How to mount

The mount command and fstab go hand in hand:

1. Options for mount and fstab are similar.
2. If a device/partition is not listed in fstab **ONLY ROOT** may mount the device/partition.
3. Users can mount a removable device using pmount.
4. Users may mount a device/partition if the device is in fstab with the proper options.

How to mount

[Mount Partitions Automatically](#) (At BOOT).

[Filesystems and Mounting](#) **Thanks Hermanzone**

mount has a multitude of options. Manpage: [man mount](#)

pmount: Pmount allows a user to mount removable media.
pmount uses /media/<NAME> as the mount point.

Syntax:

```
pmount <device> <NAME>
```

Example:

Code:

```
pmount /dev/dsa1 data
```

This creates a directory "data" in /media (mount point is /media/data) and mounts your removable device there.

To unmount:

Code:

```
pumount <NAME>
```

Note: pmount does not like to mount to an existing directory in /media.

- For example, if you have a directory /media/usb ; pmount /dev/sda1 usb may fail.
- If you are having problems with gnome-volume-manager or pmount check the contents of /media and delete directories as needed.
- Obviously do not delete a directory in /media if a device is mounted to this mount point.

[Configure pmount for internal drives](#)

To show your partitions/usb devices, first plug in your usb card.

To list your mounted partitions:

Code:

```
mount
```

To list all your partitions, mounted or not:

Code:

```
sudo fdisk -l
```

To list all your partitions by **UUID**:

First connect all your devices, then:

Code:

```
ls /dev/disk/by-uuid -alh
```

fstab Syntax

```
[Device] [Mount Point] [File_system] [Options] [dump] [fsck order]
```

Device = Physical location.

/dev/hdxy or /dev/sdxy.

x will be a letter starting with a, then b,c,....

y will be a number starting with 1, then 2,3,....

Thus hda1 = First partition on the master HD.

See [Basic partitioning](#) for more information

Note: zip discs are always numbered "4".

Example: USB Zip = /dev/sda4.

Note: You can also identify a device by udev, volume label (AKA LABEL), or uuid.

These fstab techniques are helpful for removable media because the device (/dev/sdxy) may change. For example, sometimes the USB device will be assigned /dev/sda1, other times /dev/sdb1. This depends on what order you connect USB devices, and where (which USB slot) you use to connect. This can be a major aggravation as you must identify the device before you can mount it. fstab does not work well if the device name keeps changing.

To list your devices, first put connect your USB device (it does not need to be mounted).

By volume label:

Code:

```
ls /dev/disk/by-label -lah
```

By id:

Code:

```
ls /dev/disk/by-id -lah
```

By uuid:

Code:

```
ls /dev/disk/by-uuid -lah
```

IMO, LABEL is easiest to use as you can set a label and it is human readable.

The format to use instead of the device name in the fstab file is:

LABEL=<label> (Where <label> is the volume label name, ex. "data").

UUID=<uuid> (Where <uuid> is some alphanumeric (hex) like fab05680-eb08-4420-959a-ff915cdfcb44).

Again, IMO, using a label has a strong advantage with **removable media** (flash drives).

See [How to use Labels](#) below.

For udev: udev does the same thing as LABEL, but I find it more complicated.

See [How to udev](#) for a very nice how to on udev.

Mount point.

This is where the partition is mounted or accessed within the "tree" (ie /mnt/hda1).

You can use any name you like.

In general

1. /mnt Typically used for fixed hard drives HD/SCSI.
2. /media Typically used for removable media (CD/DVD/USB/Zip).

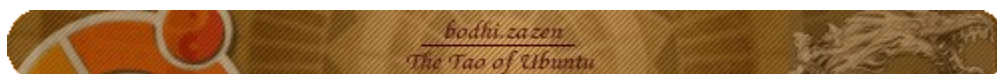
Examples:

1. /mnt/windows
2. /mnt/data
3. /media/usb

To make a mount point:

Code:

```
sudo mkdir /media/usb
```



File types:

auto: The file system type (ext3, iso9660, etc) it detected automatically. Usually works. Used for removable devices (CD/DVD, Floppy drives, or USB/Flash drives) as the file system may vary on these devices.

Linux file systems: ext2, ext3, jfs, reiserfs, reiser4, xfs, swap.

Windows:

vfat = FAT 32, FAT 16

ntfs= NTFS

Note: For NTFS rw [ntfs-3g](#)

CD/DVD/iso: iso9660

To mount an iso image (*.iso NOT CD/DVD device):

Code:

```
sudo mount -t iso9660 -o ro,loop=/dev/loop0 <ISO_File> <Mount_Point>
```

Network file systems: **This section assumes the server and client are already setup.**

nfs Example:

```
server:/shared_directory /mnt/nfs nfs <options> 0 0
```

More detailed information on [nfs](#)

smb (samba) : Samba mounts can be performed very easily via gui tools (See [Ubuntu Wiki Setting up Samba](#)). If you mount a samba share with the gui tools it will be placed in ~/.gvfs , a hidden directory in your home directory.

This section is limited to fstab and you will need a fstab entry to mount samba shares at boot.

smbfs is now depreciated for cifs : <http://linux-cifs.samba.org/>

cifs still uses a credentials file to avoid the need to enter a password. If you do not use a credentials file, you will mount a samba share with sudo and enter your username and password in a terminal.

```
//Server/share /mnt/samba cifs users,auto,credentials=/path/credentials_file,noexec 0 0
```

- Server = Name (if in /etc/hosts) or IP Address of samba server.
- share = Name of shared directory (folder).
- /mnt/samba = your desired mount point.
- /path/credentials_file = full path to your credentials file. A credentials file should be owned by root (permissions 400) and contain two lines :

```
username = samba_user
password = samba_user_password
```

samba_user = samba user (on server).

samba_user_password = samba user password (on server).

- noexec for security (it can be bypassed ...).

smbfs : depreciated, but similar.

```
//win_box/shared_folder /mnt/samba smbfs rw,credentials=/home/user_name/winbox-credentials.txt 0 0
```

And from Buck2348:

Quote:

I don't mount any vfat shares but uid and gid work with smbfs shares. I might have to try out your syntax.

I could not automount at boot my smbfs shares until I found the [this fix](#) in the Forums. I hope it will help someone else. I think the problem was related to the fact that I don't use a username and password in the Windows systems. All I had to do was add

Code:

```
username=share,password=
```

to the options list in the fstab line for these shares.

More detailed information on see : [The Official Samba 3.2.x HOWTO and Reference Guide](#)

[sshfs](#) : Network shares over ssh

<http://ubuntu.wordpress.com/2005/10/28/how-to-mount-a-remote-ssh-filesystem-using-sshfs/>

Code:

```
sshfs#user@server:/share fuse user,allow_other 0 0
```

- "Server" = Samba server (by IP or name if you have an entry for the server in your hosts file)
- "share" = name of the shared directory

Options:

Ubuntu 8.04 now defaults to "relatime". For a discussion of this option see : <http://lwn.net/Articles/244829/>

defaults = rw, suid, dev, exec, auto, nouser, and async.

Options for a separate /home : nodev,nosuid,relatime

My recommended options for removable (USB) drives are in green.

auto= mounted at boot

noauto= not mounted at boot

user= when mounted the mount point is owned by the user who mounted the partition

users= when mounted the mount point is owned by the user who mounted the partition and the group users

ro= read only

rw= read/write

VFAT/NTFS:

Ownership and permissions of vfat / ntfs are set at the time of mounting. This is often a source of confusion.

uid= Sets owner. Syntax: may use user_name or user ID #.

gid= sets group ownership of mount point. Again may use group_name or GID #.

umask can be used to set permissions if you wish to change the default.

Syntax is "odd" at first.

To set a permissions of 777, umask=000

To set permissions of 700, umask=077

Best is to set directories with executable permissions and file with read write. To do this, use fmask and dmask (rather than umask):

dmask=027

fmask=137

With these options files are not executable (all colored green in a terminal w/ ls)

Linux native file systems: Use defaults or users. To change ownership and permissions, mount the partition, then use chown and chmod.

Note: Warning re: sync and flash devices:

Warning

Additional Options: (From wiki.linuxquestions.org/wiki/Fstab):

- sync/async - All I/O to the file system should be done (a)synchronously.
- auto - The filesystem can be mounted automatically (at bootup, or when mount is passed the -a option). This is really unnecessary as this is the default action of mount -a anyway.
- noauto - The filesystem will NOT be automatically mounted at startup, or when mount passed -a. You must explicitly mount the filesystem.
- dev/nodev - Interpret/Do not interpret character or block special devices on the file system.
- exec / noexec - Permit/Prevent the execution of binaries from the filesystem.
- suid/nosuid - Permit/Block the operation of suid, and sgid bits.
- ro - Mount read-only.
- rw - Mount read-write.
- user - Permit any user to mount the filesystem. This automatically implies noexec, nosuid,nodev unless overridden.
- nouser - Only permit root to mount the filesystem. This is also a default setting.
- defaults - Use default settings. Equivalent to rw, suid, dev, exec, auto, nouser, async.
- _netdev - this is a network device, mount it after bringing up the network. Only valid with fstype nfs.

Dump

Dump: Dump field sets whether the backup utility dump will backup file system. If set to "0" file system ignored, "1" file system is backed up.

Fsck order

Fsck: Fsck order is to tell fsck what order to check the file systems, if set to "0" file system is ignored.

See also: [Tuning the Filesystem Check at Bootup](#)

Fstab Examples

```

/dev/sda14 /mnt/zen ext3 relatime 0 2

# Usb device (assuming vfat)
/dev/sdb1 /media/usb auto users,uid=1000,gid=100,utf8,dmask=027,fmask=137 0 0

#Data partition
LABEL=data /mnt/usr_data ext3 auto,users,rw,relatime 0 0

# Flash drive By UUID
UUID=fab05680-eb08-4420-959a-ff915cdfcb44 /media/flash vfat user,uid=1000,gid=100,utf8,dmask=027,fmask=137
0 0

/dev/disk/by-id/usb-IOMEGA_ZIP_250_059B00301400B0F1-part4 /mnt/zip vfat
users,uid=1000,gid=100,utf8,dmask=027,fmask=137 0 0

/dev/hda1 /mnt/windows ntfs-3g auto,users,uid=1000,gid=100,utf8,dmask=027,fmask=1 37 0 0

# VFAT
# FAT ~ Linux calls FAT file systems vfat)
# /dev/hda1
UUID=12102C02102CEB83 /media/windows vfat auto,users,uid=1000,gid=100,utf8,dmask=027,fmask=1 37 0 0

# NTFS ~ Use ntfs-3g for write access (rw)
# /dev/hda1
UUID=12102C02102CEB83 /media/windows ntfs-3g auto,users,uid=1000,gid=100,utf8,dmask=027,fmask=1 37 0 0

# Separate Home
# /dev/sda7
UUID=413eee0c-61ff-4cb7-a299-89d12b075093 /home ext3 nodev,nosuid,relatime 0 2

# Samba
//server/share /media/samba cifs user=user,uid=1000,gid=100 0 0
# "Server" = Samba server (by IP or name if you have an entry for the server in your hosts file
# "share" = name of the shared directory
# "user" = your samba user
# This set up will ask for a password when mounting the samba share. If you do not want to enter a password, use a
credentials file.
# replace "user=user" with "credentials=/etc/samba/credentials" In the credentials file put two lines
# user=user
# password=password
# make the file owned by root and ro by root (sudo chown root.root /etc/samba/credentials && sudo chmod 400
/etc/samba/credentials)

# NFS
Server:/share /media/nfs nfs rsize=8192 and wsize=8192,noexec,nosuid
# "Server" = Samba server (by IP or name if you have an entry for the server in your hosts file
# "share" = name of the shared directory

#SSHFS
Sshfs#user@server:/share fuse user,allow_other 0 0
# "Server" = Samba server (by IP or name if you have an entry for the server in your hosts file
# "share" = name of the shared directory

```

How to Label

Linux: How the label and the UUID are set depends on the file system type used. It can normally be set when creating/formatting the file system and the file system type usually has some tool to change it later on (e.g. e2tune2fs, xfs_admin, reiserfstune, etc.)

Labels

Mke2fs/e2label/tune2fs:

Note: For either ext2 or ext3 file systems.

WARNING: mke2fs will reformat your partition and set a label at the same time. This will delete any data on the target partition.

To set a label without reformatting use e2label or tune2fs

1. Make a label:

Code:

```
mke2fs -L <label> <dev>
```

OR

Code:

```
e2label <dev> <label>
```

OR

Code:

```
tune2fs -L <label> <dev>
```

Examples:

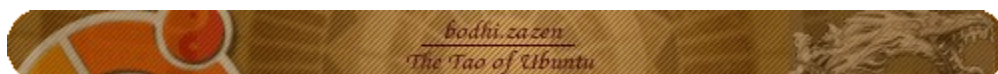
```
mke2fs -L data /dev/hda3
```

OR

```
e2label /dev/hda3 data
```

OR

```
tune2fs -L data /dev/hda3
```



2. Create a mount point:

```
sudo mkdir /media/data
```

3. Add an entry to /etc/fstab:

```
LABEL=data /media/data ext3 defaults 0 0
```

4. To mount:

```
sudo mount LABEL=data
```

ReiserFS:

Use reiserfstune:

Code:

```
reiserfstune --l <Label> <device>
```

Note: That is a small "L" and not the number 1.

JFS:

Use jfs_tune:

Code:

```
jfs_tune -L <Label> <device>
```

To show the label:

Code:

```
jfs_tune -l <device>
```

Note: That is a small "L" and not the number 1.

XFS:

Use xfs_admin:

Code:

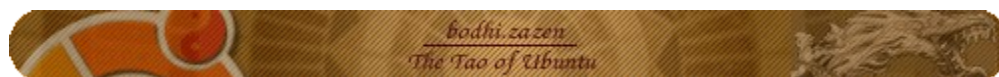
```
sudo xfs_admin -L <Label> <device>
```

To show the label:

Code:

```
xfs_admin -l <device>
```

Note: That is a small "L" and not the number 1.



FAT (Windows partitions):

Use mtools to label a FAT partition:

1. Install mtools:

Code:

```
sudo aptitude install mtools
```

2. Copy the mtools configuration file to ~:

Code:

```
cp /etc/mtools.conf ~/.mtoolsrc
```

Note: ~ is shorthand for /home/user_name.

3. Mount your flash drive.

4. Edit ~/.mtoolsrc:

Code:

```
gedit ~/.mtoolsrc
```

5. Add these lines to the end of ~/.mtoolsrc:

```
drive i: file="<device>"  
mtools_skip_check=1
```

Where <device> is the device assigned to your mounted USB device/flash drive (ie sda1, sdb1, ...).

Note: You can do this from the command line:

Code:

```
echo 'drive i: file="<device>"' >> ~/.mtoolsrc  
echo mtools_skip_check=1 >> ~/.mtoolsrc
```

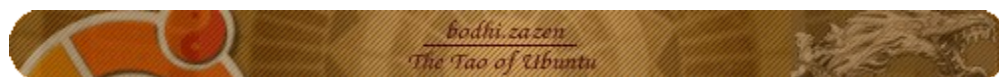
Although you will need to edit ~/.mtoolsrc for each new device if the device assignment changes.

Example: = drive i: file="/dev/sda1"

6. Change to drive i:

Code:

```
mcd i:
```



7. Check the current label:

Code:

```
mlabel -s i:
```

8. Change the current label:

Code:

```
sudo mlabel -s i:DATA
```

9. Or

Code:

```
sudo mlabel i:DATA
```

pieroxy reports the -s flag did not work, [thanks pieroxy](#)

Note: mlabel USES ALL CAPS.

10. Add an entry to fstab:

```
LABEL=DATA <mount_point> vfat defaults 0 0
```

Note: You can also mount the usb device with:

Code:

```
mount LABEL=<label>
```

NTFS (Windows partitions):

[Thanks to rudyj for pointing out the oversight.](#)

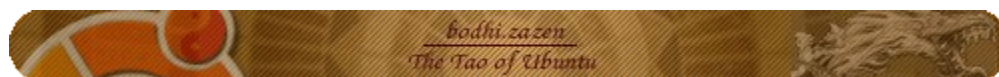
Use ntfsprogs:

First install ntfsprogs:

Code:

```
sudo aptitude install ntfsprogs
```

Or use Synaptic.



Then:

1. Show label:

Code:

```
ntfslabel <device>
```

2. Change label:

Code:

```
ntfslabel <device> <label>
```

Where:

- **<label> = your new label**
- **<device> = your partition to label (/dev/hda1 perhaps)**

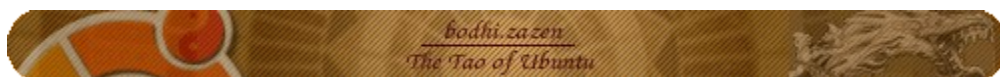
3. Add an entry to fstab:

```
LABEL=DATA <mount_point> ntfs(or ntfs-3g) defaults 0 0
```

Note: You can also mount the usb device with:

Code:

```
mount LABEL=<label>
```



Examples of fstab options

***** FAT *****

FAT partitions are easy to share between Linux and Windows as both OS will read FAT "out of the box" without additional installation or configuration.

In this example I will use /mnt/data as my mount point.

Code:

```
sudo mkdir /mnt/data
```

fstab:

```
LABEL=data /mnt/data vfat <see options below> 0 0
```

Default permissions of /mnt/data:

```
drwxr-xr-x 2 root root
```

1. fstab options: **defaults**

mount /mnt/data yields: mount: only root can mount /dev/sdb1 on /mnt/data

sudo mount /mnt/data mounts the device.

Permissions:

```
drwxr-xr-x 7 root root
```

Note: ONLY ROOT has rw permissions. 😊

1. fstab options: **users,noauto,rw**

mount /mnt/data mounts the partition.

Permissions:

```
drwxr-xr-x 7 bodhi adm
```

Note: The user can mount the device and has rw permissions. 😊

Note: The ownership and permissions of the mount point have changed !

1. fstab options: **users,noauto,gid=100,umask=007**

mount /mnt/data mounts the partition.

Permissions:

```
drwxrwx--- 7 bodhi users
```

Note: The user can mount the device and now both the user and the users group have rw permissions.

Note: The ownership and permissions of the mount point have changed again ! 😊

***** Linux Native File Systems *****

In this example I will use ext3, but this holds true for ext2, reiserfs, jfs, and xfs.

Code:

```
sudo mkdir /mnt/ext3
```

fstab:

```
LABEL=ext3 /mnt/ext3 auto <see options below> 0 0
```

1. fstab options: **defaults**

mount /mnt/data yields: mount: only root can mount LABEL=ext3 on /mnt/ext3

sudo mount /mnt/ext3 mounts the device.

Permissions:

```
bodhi@Arch:~$ls -l /mnt | grep ext3
drwxr-xr-x 3 bodhi users 1024 2006-11-07 17:26 ext3
```

Note: Ownership has changed ! owner=bodhi, group=users, however ONLY USER (and root of course) has rw permissions.

1. fstab options: **users,noauto**

mount /mnt/ext3 mounts the partition.

Permissions:

```
bodhi@Arch:~$mount /mnt/ext3/
bodhi@Arch:~$ls -l /mnt | grep ext3
drwxr-xr-x 3 bodhi users 1024 2006-11-07 17:26 ext3
```

Note: The user can mount the device and has rw permissions. 😊

Note: Ownership remains bodhi:users

Note: ext2 and ext3 do not take uid=xxx, gid=xxx, or umask=xxx

To set group rw permissions:

fstab options: **users,noauto**

1. mount the partition: **mount /mnt/ext3**
2. Set permissions of the mount point: **chmod 777 /mnt/ext3**

The set ownership and permissions will remain in effect with un-mount and re-boot.

Example:

```

bodhi@Arch:~$chmod 777 /mnt/ext3
bodhi@Arch:~$ls -l /mnt | grep ext3
drwxrwxrwx 3 bodhi users 1024 2006-11-07 17:51 ext3
bodhi@Arch:~$umount /mnt/ext3/
bodhi@Arch:~$ls -l /mnt | grep ext3
drwxr-xr-x 2 root root 4096 2006-11-07 17:28 ext3
bodhi@Arch:~$mount /mnt/ext3/
bodhi@Arch:~$ls -l /mnt | grep ext3
drwxrwxrwx 3 bodhi users 1024 2006-11-07 17:51 ext3
bodhi@Arch:~$

```

Note: The permissions revert when the partition is un-mounted **RED**

Note: The permissions remain rw when the partition is re-mounted **BLUE**

Permissions:

Note: The user can mount the device and now both the user and the users group have rw permissions. 😊

References

Partitioning: [Basic partitioning](#)

Zip dirve how-to: [How to Zip Drive](#)

Mount:

[How to mount filesystems in Linux](#)

[Ubuntu Automatically Mount Partitions](#)

[man mount](#)

[Mount Other Filesystems](#)

nfs:

[How to set up NFS](#)

[How to NFS v4](#)

[Debian/Ubuntu NFS Guide](#) Short but sweeet !

Mount Windows Sares: [Mount Windows shares permanently](#)

Fstab:

[fstab wiki](#)

[How to edit and understand /etc/fstab](#)

[Tuning the Filesystem Check at Bootup](#)

Samba:

[Setting up Samba](#)

[How to mount smbfs shares permanently](#)

Labels: [How to use Labels](#)

udev: [How to udev](#)

NTFS: [ntfs-3g](#)

[bodhi.zazen](#)

A person with ubuntu is open and available to others, affirming of others, does not feel threatened that others are able and good, for he or she has a proper self-assurance that comes from knowing that he or she belongs in a greater whole and is diminished when others are humiliated or diminished, when others are tortured or oppressed. ~

Archbishop Desmond Tutu, 1999