

### AUTO MOUNTING NFS SHARES IN /ETC/FSTAB

To mount an NFS share using fstab (/etc/fstab) you need to know a few things, the hostname or IP address of the NFS server, the share name and where you intend to mount the share at. Next, add this line to the end of the /etc/fstab file:

```
hostname_or_ip:/share_name /yourmount_location nfs
rsize=8192,wsiz=8192,timeo=14,intr 0 0
```

```
10.0.0.50:/backup /nfs/backup nfs
rsize=8192,wsiz=8192,timeo=14,intr 0 0
```

When you are done, you can quickly refresh the fstab using the mount command

### fstab Options common to all file-systems

auto / noauto With the auto option, the device will be mounted automatically at bootup or when the mount -a command is issued. auto is the default option. For the device not to be

dev / nodev Interpret/do not interpret block special devices on the file-system.

exec / noexec exec lets binaries that are on the partition be executed, whereas noexec is the opposite.

rw / ro Mount the file-system in either read write or read only mode.

sync / async How the input and output to the filesystem should be done. sync means it is done synchronously.

suid / nosuid Permit/Block the operation of suid, and sgid bits.

### fstab Options common to all file-systems (cont)

user / users / nouser user permits any user to mount the filesystem. This automatically implies noexec, nosuid, nodev unless overridden. If nouser is specified, only root can mount the filesystem. If users is specified, every user in group users will be able to unmount the volume.

defaults Use default settings. Default settings are defined per file system at the file system level. The normal default for Ext3 file systems is (rw,suid,dev,exec,auto,nouser,async)

owner (Linux-specific) Permit the owner of device to mount.

atime / noatime / relatime / strictatime (Linux-specific) The Unix stat structure records when files are last accessed (atime), modified (mtime), and changed (ctime). One result is that atime is written every time a file is read, which has been heavily criticized for causing performance degradation and increased wear.

### File-system-specific options

ext2	check={n one, normal, strict}	Sets the fsck checking level.
	debug	Print debugging info on each remount .
	sb=n	n is the block which should be used as the superblock for the fs.
fat	check={r [elaxed], n[ormal], s[trict]}	Not the same as ext2, but rather deals with allowed filenames. See mount.
	conv={b[inary], t[ext], a[uto]}	Performs DOS <---> UNIX text file conversions automatically. See mount.
fat, ntfs	windows _names	Linux filesystems have a larger set of allowed characters in filenames. windows_names restricts the set of allowed characters for the volume to only those acceptable by Windows. Note: though FAT/NTFS are the most common use case, this feature is not specifically restricted to those filesystem types.
	uid=n, gid=n	Sets the user identifier, uid, and group identifier, gid, for all files on the filesystem.
	umask=nnn, dmask=nnn, fmask=nnn	
nfs	addr=ip	where 'ip' means IP address

### nfs - fstab format and options for nfs

nfsvers=n		The NFS protocol version number used to contact the server's NFS service.
soft / hard		Determines the recovery behavior of the NFS client after an NFS request times out.
intr / nointr		This option is provided for backward compatibility. It is ignored after kernel 2.6.25.
timeo=n		The time in deciseconds (tenths of a second) the NFS client waits for a response before it retries an NFS request. For NFS over TCP the default timeo value is 600 (60 seconds).
retrans=n		The number of times the NFS client retries a request before it attempts further recovery action. If the retrans option is not specified, the NFS client tries each request three times.
rsize=n		The maximum number of bytes in each network READ request that the NFS client can receive when reading data from a file on an NFS server.
wsize=n		The maximum number of bytes per network WRITE request that the NFS client can send when writing data to a file on an NFS server.
actimeo=n		Using actimeo sets all of acregmin, acregmax, acdirmin, and acdirmax to the same value.
bg / fg		Determines how the mount command behaves if an attempt to mount an export fails.



### nfs - fstab format and options for nfs (cont)

**retry=n** The number of minutes that the mount command retries an NFS mount operation in the foreground or background before giving up.

#### # Examples

```
server:path /mountpoint fstype option,option,... 0 0
```

The following example from an `/etc/fstab` file causes the mount command to negotiate reasonable defaults for NFS behavior.

```
server:/export /mnt nfs defaults 0 0
```

This example can be used to mount `/usr` over NFS.

```
server:/export /usr nfs ro,nolock,nocto,actimeo=3600 0 0
```

additional info:

<https://www.systutorials.com/docs/linux/man/5-nfs/>

### Configuration Options NFS side

**rw** This option gives the client computer both read and write access to the volume.

**sync** This option forces NFS to write changes to disk before replying. This results in a more stable and consistent environment since the reply reflects the actual state of the remote volume. However, it also reduces the speed of file operations.

**no\_subtree\_check** This option prevents subtree checking, which is a process where the host must check whether the file is actually still available in the exported tree for every request. This can cause many problems when a file is renamed while the client has it opened. In almost all cases, it is better to disable subtree checking.

### Configuration Options NFS side (cont)

**no\_root\_squash** By default, NFS translates requests from a root user remotely into a non-privileged user on the server. This was intended as a security feature to prevent a root account on the client from using the file system of the host as root. `no_root_squash` disables this behavior for certain shares.

example:

in the file `/etc/export`

```
# the location | the network | the options |
/mount/folder 192.168.1.0/24(rw,sync,no_root_squash)
```

### Sample NFS fstab entry

To mount an export using NFS version 2, use the `nfs` file system type and specify the `nfsvers=2` mount option. To mount using NFS version 3, use the `nfs` file system type and specify the `nfsvers=3` mount option. To mount using NFS version 4, use either the `nfs` file system type, with the `nfsvers=4` mount option, or the `nfs4` file system type. The following example from an `/etc/fstab` file causes the mount command to negotiate reasonable defaults for NFS behavior.

A sample `fstab` entry for NFS share is as follows

```
server:/export /mnt nfs defaults 0 0
```

```
host.myserver.com:/home /mnt/home nfs
```

```
rw,hard,intr,rsize=8192,wsiz=8192,timeo=14 0 0
```

