

# Network File System Notes

Ravi Kumar.M.S

January 29, 2004



[Introduction](#)

[Understanding NFS](#)

[NFS Advantages](#)

[NFS Disadvantages](#)

[NFS Configuration](#)

[NFS Server Daemons](#)

[Configuring NFS...](#)

[Home Page](#)

[Print](#)

[Title Page](#)



[Page 1 of 8](#)

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## 1. Introduction

Network File System is the most common method for providing file sharing services on Linux and Unix networks. It is a distributed file system that enables local access to remote disks and file systems. NFS is also popular file sharing protocol, so NFS clients are available for many non-Unix Operating Systems, including Windows, MacOS, etc.

## 2. Understanding NFS

NFS uses standard client/server architecture. The server portion consists of disks containing shared file systems that are visible to client systems on the network. This process by which a server exports its file systems to clients is called *exporting a file system*. Server daemons also provide services such as file locking, quota management on NFS exports. NFS clients simply mount the exported file systems (NFS mounts), on their local systems just as they would mount file systems on their local disks.

## 3. NFS Advantages

Clearly, the biggest advantage of NFS is centralised administration. It is much easier, for example, to backup a file system stored on a server than to

individually back up a scattered system. NFS when used with NIS makes it trivially simple to update key configuration files. NFS can also conserve disk space and prevent duplication of resources. Likewise, upgrading applications employed by users throughout a network simply becomes a matter of installing the new application and changing the exported file system to point at the new application. End users also benefit from NFS. When NFS is combined with NIS, users can log in from any system, even remotely, and still have access to their home directories and see a uniform view of shared data.

## 4. NFS Disadvantages

NFS has its shortcomings, primarily in terms of performance and security. As a distributed, network based filesystem; NFS is sensitive to network congestion. Heavy disk activity of the NFS server adversely affects the NFS's performance. If an exported file system is not available when the client attempts to mount, the client system hangs, although this can be mitigated using a specific mount. If the server hosting the exported file system becomes unavailable due to any reason, no one can access the resource. NFS has security problems because its design assumes a trusted network.

## 5. NFS Configuration

Designing a NFS server involves ...

1. Selecting the filesystem to export.
2. Choosing which users (clients) are permitted to mount the exported filesystems.
3. Selecting a naming convention and mounting scheme that maintains network transparency and ease of use.
4. Choosing the server and client systems to follow the convention.

### 5.1. Configuring NFS Server

Configuration and status files	Daemons	Scripts and Commands
/etc/exports	rpc.portmap	/etc/rc.d/init.d/nfs
/var/lib/nfs/rmtab	rpc.mountd	nfsstat
/var/lib/nfs/xtab	rpc.nfsd	showmount
/etc/hosts.allow	rpc.statd	rpcinfo
/etc/hosts.deny	rpc.lockd	exportfs

The server configuration file is `/etc/exports`, which contains a list of file systems to export, the clients permitted to mount them, and the export options that apply to client mounts. Each line in the `/etc/exports` has

[Home Page](#)[Print](#)[Title Page](#)[Page 5 of 8](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

the following format:

```
dir host(options) [host(options)] ...
```

...where **dir** is a directory or file system to export, **host** specifies one or more hosts permitted to mount **dir**, and **options** specifies one or more mount options, hosts can be specified as a single name, an NIS group, as a group of hosts using the wild card characters '?' and '\*'. When specified as a single name, **host** can be any name that DNS can resolve into an IP address. If host is an NIS netgroup, it is specified as @groupname.

Consider a sample `/etc/exports` file :

```
/usr/local *.pes.edu
/home 192.168.0.0/255.255.255.0 (rw)
/projects @deeproot(rw)
/var/tmp 192.168.0.1(rw)
```

The first line permits all hosts with a name of format *somehost.pes.edu* to mount `/usr/local`. The second line permits any host with an IP address in the range 192.168.0.0 to 192.168.0.1 to mount `/home`. The third line permits any member of the NIS netgroup named `deeproot` to mount `/projects`. The final line permits only the host whose IP address is 192.168.0.1 to mount `/var/tmp`.

Find more about the `/etc/exports` file in 'man exports'.

Two additional files store status information about NFS exports, They are `/var/lib/nfs/rmtab` and `/var/lib/nfs/xtab`. Each time the `rpc.mountd`

daemon, which services mount requests for exported file system receives a mount request, it adds an entry to `/var/lib/nfs/rmtab`.

## 6. NFS Server Daemons

Assuming that you have configured your `/etc/exports` file as shown in the previous table, the next step is to start the daemons. This is done as follows:

```
# /etc/rc.d/init.d/nfs start
# /etc/rc.d/init.d/nfslock start
```

Next step is to make sure the necessary daemons are running :

```
# rpcinfo -p
```

Now list and view the server's NFS exports :

```
# showmount -a
```

Lastly, export the filesystem :

```
# exportfs -v
```

DAEMON	FUNCTION
portmap	Enables NFS clients to discover NFS services available on a given server.
mountd	Processes NFS client mount requests.
nfsd	Provides all NFS services except file locking and file management.
lockd	Starts the kernel's NFS lock manager
statd	Implements NFS lock recovery when an NFS server system crashes.
rquotad	Provides file system quota information about NFS exports to NFS clients.

## 7. Configuring NFS Client

Configuring client systems to use NFS involves making sure that the portmapper and NFS file locking daemons `statd` and `lockd` are available, adding entries to the `/etc/fstab` file for NFS exports, and mounting the exports using `mount` command. Make sure that the portmapper is running on the client system using the portmap initialization script `/etc/rc.d/init.d/portmap`. In order to use NFS file locking, both NFS server and any NFS clients need to run `statd` and `lockd`.

[Introduction](#)

[Understanding NFS](#)

[NFS Advantages](#)

[NFS Disadvantages](#)

[NFS Configuration](#)

[NFS Server Daemons](#)

[Configuring NFS...](#)

[Home Page](#)

[Print](#)

[Title Page](#)

◀

▶

◀

▶

Page 7 of 8

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

[Home Page](#)[Print](#)[Title Page](#)[Page 8 of 8](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

To mount the file system, use the following command :

```
# mount -t nfs host:/home /home
      rsize=8292,wsiz=8192,hard,intr,nolock
```

Clients need to have the following entries in their `/etc/fstab` file :

```
host:/usr/local /local nfs \
      rsize=8292,wsiz=8192, hard,intr,nolock 0 0
host:/home      /home  nfs  \
      rsize=8292,wsiz=8192, hard,intr,nolock 0 0
```

Now start the *portmapper*:

```
# /etc/rc.d/init.d/portmap start
```

Mount the exports using one of the following commands :

```
# mount -a -t nfs
```

Or

```
mount /home /usr/local
```

The first command mounts all (-a) directories of type nfs, while the second mounts only the specified file systems.

---