

# How to Set Up NFS Server Support on a Cobalt NASRaQ

## Table of Contents

<b>Part I.</b>	<b>Enabling and Disabling NFS</b>
<b>Part II.</b>	<b>Configuring NFS</b>
<b>Part III.</b>	<b>Options</b>
<b>Part IV.</b>	<b>Users and Groups</b>
<b>Part V.</b>	<b>NASRaQ Share Points</b>
<b>Part VI.</b>	<b>Examples</b>
<b>Part VII.</b>	<b>Security Concerns</b>
<b>Part VIII.</b>	<b>Where to Get More Information</b>

## Overview

This document is targeted for users of the NASRaQ desiring NFS support, who has an understanding for Unix or Linux.

The NASRaQ does not provide NFS server support via the web user interface. As a result, you can provide NFS services from your NASRaQ using the command-line interface, which could potentially void your warranty.

The following instructions assume basic familiarity with connecting to a NASRaQ via the telnet protocol, using the command-line interface, and editing configuration files through the use of a text editor.

NFS server support involves several steps:

- 1) enabling the service
- 2) editing a configuration file to suit your needs
- 3) telling your NFS server about changes in the configuration file
- 4) disabling the service, if desired.

The first part of this document provides a walk-through of how to perform these steps on the Linux-based operating system used by the NASRaQ. Following that, a more detailed overview of the commands and options available will be given.

## Part I. Enabling and Disabling NFS

An NFS server consists of two parts: a portmap program that receives NFS requests and the NFS server programs that actually handle the request. To start up NFS, then, you need to start both services. To do so, enter the following commands:

- 1) `/etc/rc.d/init.d/portmap start`
- 2) `/etc/rc.d/init.d/nfs start`

Similarly, the following commands will install the necessary support needed to enable NFS server support at system startup:

- 1) `/sbin/chkconfig --add portmap`
- 2) `/sbin/chkconfig --add nfs`

To disable NFS server support while your machine is running, perform the following commands:

- 1) `/etc/rc.d/init.d/nfs stop`
- 2) `/etc/rc.d/init.d/portmap stop`

Finally, to prevent NFS server support from being enabled at startup, use the following commands:

- 1) `/sbin/chkconfig --del portmap`
- 2) `/sbin/chkconfig --del nfs`

## Part II. Configuring NFS

Enabling or disabling access to share points involves two steps:

- 1) editing `/etc/exports`
- 2) `/usr/sbin/exportfs -a`

Share points are the file locations for sharing purposes. The first step configures your share points for use by other machines while the second tells the NFS server to re-read the configuration file.

**NOTE:** The format of the `exports` file is different from other non-Linux Unix platforms with which you may have some familiarity.

The `/etc/exports` file consists of lines of the following form:

```
path machine(options),machine2(options)
```

Lines that begin with the '#' character are considered comments. The 'path' argument consists of the Unix path for the share point desired. 'Machine' can be either the actual client machine name or an IP address (e.g., myhost or 192.168.0.100). The client machine name can contain Unix wildcard characters (e.g., \*.domain.net) while the IP address can be given in the address/netmask form (e.g., 192.168.0.0/255.255.255.0). 'Options' consist of a comma-separated list of directives (e.g., `ro` or `no_root_squash`) that alter the behavior of the NFS server to its clients. Usually, they are used to restrict access rights to a particular share point.

To check to see if the NFS server has re-read the new configuration correctly, user `'/usr/sbin/showmount -e.'`

### Part III. Options

Following is a list of some commonly used NFS options:

ro	Grant read-only access to the share point
rw	Grant read-write access to the share point (default)
root_squash	Map the root user and to the anonymous user and group. This is given to prevent root access by unknown client machines posing as root. (default)
no_root_squash	Allow the root user to retain root access rights.
squash_uids=A-B,C,D	Map the specified user ids to the anonymous user. In this case, user id's A through B, C, and D will get mapped to the anonymous user.
squash_gids=A-B,D-E	Map the specified group ids to the anonymous group. In this example, A through B and D through E will get mapped to the anonymous group.
all_squash	Map all users and groups to the anonymous user and group.
no_all_squash (default)	Do not map users and group to the anonymous user and group.
map_static=/path/to/file ids.	Specify a mapping between the client and server user and group

Here is the format:

# user/group	client	server	
uid	0-99	-	# map to anonymous
uid	100-500	1000	# map to 1000-1500
gid	0-49	-	# map to anonymous
gid	50-100	2000	# map to 2000-2050

Secure clients requesting NFS services must be using a port number below 1024. On Unix-based systems, port numbers below 1024 are restricted to the root user. In actuality, this option actually provides little additional security over the insecure option. (default)

Insecure clients requesting NFS services may originate from a port number greater than or equal to 1024.

## **Part IV. Users and Groups**

Like other multi-user operating systems, the Linux-based Cobalt OS uses numbers to represent the users and groups on the NASRaQ. Many of the NFS options relate to the correspondence between the user and group ids on the NFS server and the NFS client.

If you wish to have users utilize the same user and group ids, you will need to translate the NFS client's users and groups to the equivalent users and groups on the server. The simplest way to do this is to examine the `/etc/passwd` and `/etc/group` files and use the `'map_static'` option (see the Options section) to translate between client and server user and group IDs.

## Part V. NASRaQ Share Points

The NASRaQ provides share points on a group or user basis. The group shares are kept in `/shares/md0/shares`, and the user shares are kept in `/shares/md0/users`. While the user share names correspond to the actual user account names, the group shares do not. Instead they are numbers. To determine the actual share name for a group share, you will need to look in the `/visible` directory for symbolic links that refer to directories in `/shares/md0/shares`.

For convenience, you might want to refer to these symbolic links in the `/etc/exports` file instead of the actual directories in `/shares/md0/shares`. However, there are two caveats that you should be aware when doing so:

- 1) If the directory given in `/etc/exports` is actually a symbolic link, `showmount` will show the actual directory instead of the symbolic link. When you mount the share on the NFS client, however, you still need to use the value given in `/etc/exports`.
- 2) You cannot just put `/visible` in the `/etc/exports` file. Doing so will export only the symbolic links in the `/visible` directory and not the actual share points in `/shares/md0/shares`.

## Part VI. Examples

- 1) Allow read-only access by everyone to /shares/md0 and read-write access by asun.cobaltnet.com to /shares/md0/users/asun. For the latter share point, give the root user on asun.cobaltnet.com root access.

```
# this is a line with a comment
/shares/md0 (ro)
/shares/md0/users/asun asun.cobaltnet.com(rw,no_root_squash)
# this is the last line of the file
```

- 2) Use wildcards and netmasks to give the entire cobaltnet.com domain read-write access to /visible/voll and the 192.168.0 subnet read-only access to /visible/public.

```
/visible/voll *.cobaltnet.com(rw)
/visible/public 192.168.0.0/255.255.255.0(ro)
```

- 3) Give all education sites read-only access to /visible/public but force them to connect as the anonymous user and group.

```
/visible/public *.edu(ro,squash_all)
```

## Part VII. Security Concerns

While the NFS server provides restricted access to share points, it does not provide restricted access to the actual NFS server. As a result, it may be possible to contravene the NFS server's notion of security.

If you have security concerns regarding use of the NFS server, you may wish to restrict access to your NFS server using the `/etc/hosts.allow` and `/etc/hosts.deny` files. The NFS determine whether a machine has access to the NFS server, the `portmap` program first checks `/etc/hosts.allow` for hosts that are allowed, and then `/etc/hosts.deny` for hosts that are denied. Lines in these files are in the following form:

```
<service>: host
```

`<service>` in this case would be `'portmap'` while `host` would either be a host name or an ip address with optional netmask. The keyword `ALL` can be used in place of either of these to refer to either all services or all sites.

To restrict NFS access to only the 192.168.0 subnet, for example, you would enter the following in the `/etc/hosts.allow` file:

```
portmap:          192.168.0.0/255.255.255.0
```

and the following to the `/etc/hosts.deny` file:

```
portmap:          ALL
```

## Part VIII. Where to Get More Information

If you are familiar with Unix man pages, you can use either 'man <command>' or 'man -k <command keyword>' to get additional information on the commands discussed in this document.

At the end of this document, there will be a list of commands and files about which you might want to learn more information. In addition, there are a number of online references that also provide useful information. For example, the support page on <<http://www.linux.org>> has more detailed information on the Linux-based operating system that Cobalt uses.

Commands of interest (located in the /sbin or /usr/sbin directory):

```
chkconfig
exportfs
showmount
tcpd
portmap
```

Files (located in the /etc directory):

```
exports
hosts.allow
hosts.deny
```