

# using NFSManager

by Wayde Schroeder

NFS0\_icon.tiff ↵

If you've worked with NFS® (Network File System) administration in the past, you'll welcome the improvements under NeXTSTEP Release 3. Whether you're new to NFS administration or not, you'll find that NFSManager, an application that provides a graphic interface to the tasks involved, makes setting up and maintaining shared NFS directories a breeze.

What follows is a description of how to use NFSManager to export and import (mount) remote directories. In addition, you'll discover some of the old workarounds you might have used in Release 2 aren't needed any more.

## exporting directories

Before you jump in, make sure you've given some thought to what you want to accomplish. First, of course, you need to decide which directory you want to make available to the network.

Next, decide what type of access you want to grant: Will you export a directory to all computers on the network or just to a specific subset? Will there be read-only access or read/write access, and

will there be root privileges from some clients or none?

Once you've made your decisions and are ready, start up NFSManager. Two windows open: Exported Directories and Imported Directories. To export your directory, do the following:

1. Drag the Exported Directories window to a spot that will allow you to see both the well in the upper right corner and the File Viewer.
2. Use the File Viewer in the Workspace Manager to select the directory you want to share.
3. Drag the directory icon from the File Viewer to the well in the Exported Directories window of NFSManager.

NFS1\_Earth\_4.4.tiff ↵

4. If necessary, change the access options to match your needs. See ``about those access options" on page 20.
5. Click OK.

That's all there is to it-no need to edit /etc/exports or even run exportfs. If you prefer, you can export a directory by clicking the Add button in the upper right corner of the Exported Directories

window. A directory browser will appear, from which you can choose the directory to be exported, as shown in figure 1.

*figure 1: choosing a directory for export*

4.3.tiff ↪

You can use NFSManager to export a directory from a computer other than the one you're logged into. To do this, choose Export From from the main menu and then select the host name from the browser that appears. A new Exported Directories window for the selected host appears, enabling you to export directories. Be sure you know what directory you want to export from the remote computer because you'll be working blind-you won't be able to list the directories available on the remote system.

## **importing directories**

Just as with exporting directories, a little planning is in order before you begin importing remote directories. First, you need to decide which domain will hold the mount information; this determines which computers will mount the shared file system. Next, decide where the directory should be mounted (the mount point) and whether the directory should be mounted read/write or read-only.

Do you want to mount the directory in foreground (prevent the computer from booting until the

directory is mounted), or do you want to mount the directory in background (allow the computer to boot even if the first mount attempt fails)? If the server becomes unavailable and a user tries to access the directory, what action do you want taken-stop trying and return an error message or keep trying until the server responds? Finally, what should be done with setuid files-treat them normally or run them as the user that executes the file?

Once you've completed your planning, you can begin to step through the import process as follows:

1. Choose Import To from the main menu.
2. In the domain browser that appears, choose the domain that will hold the mount information. An Imported Directories window for the selected domain will appear.

4.7.tiff ↪

3. Click the Add button to bring up this panel.

4.8.tiff ↪

4. Enter the host name of the file server (the computer from which you exported the directory) and the name of the directory, and click OK. If you want, you can use the Select from NetInfo button

to choose the host name and directory from a browser instead.

4.11.tiff ↪

5. Modify the mount point and options as needed and click OK.

By default, the directory will be mounted under /Net, which also means an automatic mount (the directory won't actually be mounted until someone tries to access it).

**exploring behind the scenes**

Exporting information is now stored in NetInfo, along with the mounting information. You can find details about shared directories in the /exports NetInfo directory in the local domain of the file server, as shown in figure 2. Each entry represents an exported directory.

*figure 2: investigating exported directories*

4.15.tiff ↪

Along with the name of the exported directory is an opts property, as shown in figure 3, that contains the access options you set in the Exported Directories window of NFSManager.

Mount information is where it always was: under the /mounts NetInfo directory. Each subdirectory

here represents a single mount entry.

*figure 3: exploring the opts property for exports*

4.18.tiff ↵

The name of each directory is in the form server:directory. Other properties include the mount point (dir) and options (opts) as set in the Imported Directories window of NFSManager (see figure 4).

**living without workarounds**

If you've dealt with NFS directories before, you've probably encountered a problem that existed with certain kinds of mounts in earlier releases of NeXTSTEP: Exporting a directory and then mounting it in the same place wouldn't work. For example, let's say you export the directory /LocalApps from the server jupiter. Then you edit the root NetInfo domain to add an entry to mount jupiter:/LocalApps on /LocalApps. On jupiter, this causes an attempt to mount the directory /LocalApps on itself, and in previous releases, this attempt failed. In Release 3, mounts of this form succeed, with just a warning message similar to the following displayed on the server:

```
server:7# mount server:/LocalApps /LocalApps
mount: '/LocalApps' is a local directory, no need
to do an NFS mount
```

mount: giving up on:  
/LocalApps

Now you can create mounts in the network domains that will mount a directory on the same directory name without hanging the server.

Several methods of working around this problem were described in "techniques for using the Network File System" in volume 1, issue 1 of the *support bulletin*. One way was to rename the exported directory, as in server:/localapps /LocalApps-the /LocalApps directory on the server is renamed /localapps and then mounted on /LocalApps. Such workarounds aren't necessary in Release 3-the mount will fail only on the server, where it isn't needed, and will succeed on the clients.

Consider scrapping your current method and using NFSManager to reconfigure your NFS directories. Administering your shared file systems will be much simpler. If you need another reason to convert, here it is: A problem has been found that can cause the server to hang when copying files on the server into the mount point rather than the original directory. For example, copying files on jupiter into the mount point /LocalApps, instead of the original directory, /localapps, can cause jupiter to hang in an NFS wait condition. On Turbo systems, this problem can occur when copying even small amounts of data (500 KB); on non-Turbo systems, the problem occurs only with large amounts of data (over a megabyte).

## **simplifying NFS administration**

NFSManager is a welcome addition to the family of Manager applications, which already counted among its members UserManager, NetInfoManager, and PrintManager. You no longer need to edit flat files or work in NetInfoManager to set up and manage NFS directories. Although you can use NetInfoManager for these tasks, NFSManager is simpler and offers you the flexibility of managing NFS directories from a remote computer. Now you can administer your NFS directories with the same ease as you administer the other aspects of your NetInfo network.

## **about those access options**

The default access options export a directory to the entire network with read/write access (subject to the normal restrictions of file permissions, of course). If you want something else, make the appropriate modifications.

- If you want some computers to have read-only access to the directory, enter a host name in the text field below the Read Only Access column and click the Add button. Repeat for any other hosts.
- If you want read/write access granted to a subset of computers, enter a host name in the text field below the Read/Write Access column and click the Add button. Repeat for any others.
- If you want to grant root access to one or more computers, add their host names to the Root Access column.



- Finally, set the default access button. This determines what the access will be for any hosts not listed in the columns. How you set this button depends on the other access settings you've made:
- If you have any host names in the Read Only Access column, default access can only be set to No Access.
- If you have host names in the Read/Write Access column but none in the Read Only Access column, default access can be set to either Read Only or No Access.
- If you have no host names listed in either column, default access can be set to Read/Write or Read Only. (You can also set the default to No Access, but it's not a very useful choice.)