Configuring NFS [13]

13.1 Related NFS documentation

The following documentation contains information covered in this chapter:

- *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304
- *UNICOS Administrator Commands Reference Manual*, Cray Research publication SR–2022: `automount`(8), `biod`(8), `cnfsd`(8), `exportfs`(8), `mount`(8), `mountd`(8), `nfsd`(8), `nfsidmap`(8), and `sdaemon`(8) man pages
- *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR–2014: `exports`(5) and `fstab`(5) man pages

13.2 What is NFS?

The network file system (NFS) is a Cray Research software product that allows users to share directories and files across a network of machines.

NFS users can use standard I/O system calls, commands, and permission controls to access files from any file system. Similarly, other NFS users can make use of file systems by using remote commands from anywhere in the local network environment. You can use NFS in diverse administrative environments through the use of the ID mapping facility (see Section 13.3, page 300). By default, this facility is on in the UNICOS kernel. The user interface to NFS is transparent.

NFS uses a server/client system to provide access to files on the network. A *fileserver* is any machine that allows a portion of its local disk space to be exported (made available for mounting on a host machine). A *client* is any machine that makes a request for an exported file system. When a user issues an I/O call for a file that resides on a file system mounted by NFS, the call is transmitted to the server machine. When the server receives the request, it performs the indicated operation. In the case of read or write requests, the indicated data is returned to the client or written to disk, respectively. This processing is transparent to users, and it appears that the file resides on a disk drive that is local.

NFS client operations are separate from NFS server operations. This section describes the procedures for configuring a CRAY J90 system as an NFS client and as an NFS server.

For additional information about NFS, including information about the following topics, see the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304:

- NFS automounter (`automount`(8)(8) command), which is a program that runs on an NFS client that mounts and unmounts NFS file systems on demand. Using the automounter, NFS file systems are mounted only when users are accessing them.
- General security concerns; although UNICOS NFS is an excellent tool for sharing files between computer systems, it also makes the files on a server vulnerable to unauthorized access.
- Kerberos authentication, which can be required for NFS access to exported UNICOS file systems.

13.3 ID mapping and when it is used

In the UNICOS system, file access is controlled by checking the numeric user ID (UID) and group ID (GID) against the permissions bits for a file. These same rules apply to NFS. Therefore, in a standard NFS implementation, UNICOS NFS is designed to be used within one administrative domain, which is sometimes called a *flat administrative space*. An *administrative domain* is a set of hosts, usually managed by the same authority, in which all users share a common set of UIDs and GIDs. With NIS, a given user or group ID always refers to the same user or group within the administrative domain. This allows all hosts in the NFS group to interpret the authentication information passed in the NFS requests in the same way. Traditional NFS environments make use of NIS (formerly called Yellow Pages) to achieve a flat administrative space. NIS is a distributed look-up service that maintains a common database of UID and GID information for members of an administrative domain. An NIS domain is one administrative domain.

If your Cray Research system resides entirely within one NIS domain and does not interact with hosts outside that domain, you probably will not have to configure NFS ID mapping. However, Cray Research systems are often shared by many different administrative domains, making the creation of a single, flat administrative space for user and group identification technically and/or organizationally difficult. Because a given ID can refer to different users or groups in different administrative domains, this would prevent NFS from being used in such an environment, or would cause serious security problems.

The Cray Research system NFS ID mapping facility allows different administrative domains to participate in cross-mounting NFS file systems without creating a single, flat administrative space.

Following is a description of circumstances in which it is desirable and circumstances in which it is necessary for the Cray Research NFS server to access ID mapping information:

- Account IDs, or ACIDs, which are unique to the UNICOS system, are not passed across the network as part of the NFS protocol. If ID mapping is configured, NFS servers can use the requesting user's ACID for operations such as file creation. This allows NFS-created files to be charged correctly when using ACIDs for disk accounting and/or file quotas.
- On UNICOS MLS systems (with or without using the IP security option), a UNICOS NFS server must be able to validate requests based on the user's security levels and compartments. If you want to export and serve file systems on a UNICOS MLS system, ID mapping is required.
- If you want to export file systems by using the `-krb` option (Kerberos authentication), ID mapping is required so that the kernel has a place to put a list of authenticated addresses for each Kerberos user.

For additional information on NFS ID mapping, see the Network File System (NFS) chapter in the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304.

Procedure 40: Configuring a CRAY J90 system as an NFS client

Note: If you are the administrator on the server(s) and the client(s), you should know whether you exported the file systems you want to mount. To see the file systems that are currently exported, execute the `exportfs` command without arguments on the server.

The following steps explain how to configure a CRAY J90 system as an NFS client:

1. Make entries in the `/etc/fstab` file that describe the file systems you want mounted using NFS. You can use the menu system to do this step or you can do this manually.

   If you are using the menu system, you must first enable the menu system to configure NFS. To give the menu system permission to change the configuration for NFS, change the `NFS configuration` option to `YES` in the `Configure System->Configurator Automation Options` menu. Also, ensure that the `Configure System->Major Software`

`Configuration` menu has the `Network Information Service (NIS)` option set to `on`; if you must change the `Major Software Configuration` menu, you must rebuild your kernel.

Then, select the `Configure System->File System (fstab) Configuration->NFS File Systems` menu, add your entries, and update the form file. Then activate your changes through the `File Systems (fstab) Configuration` menu. A sample `Network File System Configuration` menu screen follows:

```
Configure System
   ->File System (fstab) Configuration
      ->NFS File Systems
```

```
    Network File System Configuration
      Host    Name             Mount         RW  Quota  Suid  Auto  Bg  So
     -------  --------         ------------  --  -----  ----  ----  --  -- >
 E-> tngmoon  /usr/bin         /UTNA/sunbin  ro                     bg  so
```

If you are not using the menu system, edit the `/etc/config/rcoptions` file and set the `RC_NFS=` parameter to `YES`. Then make entries in the `/etc/fstab` file that describe the file systems you want mounted using NFS. The following example shows sample entries; for more information about the options, see the `fstab`(5) and `mount`(8) man pages and *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304:

```
# cat fstab
#
# Mainframe file system table (fstab)
#
#       There are six fields per line, separated by white space.
#       1.  device name
#       2.  filesystem name
#       3.  filesystem type
#       4.  mount options
#       5.  dump frequency
#       6.  pass number to check file system
#
/dev/dsk/root   /                    NC1FS  rw    1    1
/dev/dsk/home   /home                NC1FS  rw    1    2
/dev/dsk/core   /core                NC1FS  rw    1    2
/dev/dsk/usr    /usr                 NC1FS  rw    1    2
/dev/dsk/src    /usr/src             NC1FS  rw    1    2
#
# NFS file systems
#
tngmoon:/home/tngmoon/user1     /UTNA/user1     NFS     ro,soft,bg
tngmoon:/usr/bin                /UTNA/sunbin    NFS     ro,soft,bg
```

2. Start the `biod` daemon, which is an optional client daemon that handles write-behind and read-ahead requests. Although this daemon is optional, you should run it to improve NFS performance. By default, four `biod` daemons are started; you might improve client performance by running more `biod` daemons. Ensure that the `biod` daemon is started by using the menu system or by doing it manually.

   Note: To configure NFS, the `portmap` daemon must be running (that is, it must be set to `YES` in the `/etc/config/daemons` file); `portmap` is part of the TCP daemons group. For information about starting system daemons, see Chapter 4, page 45.

   If you are using the menu system, select the `Configure System->System Daemons Configuration->System Daemons Table` menu, set the `biod` daemon to `YES`, and update the form file. Then activate your change through the `System Daemons Configuration` menu. When you activate this change, the `biod` daemon will be started automatically each time you start the UNICOS system. A sample `System Daemons Table` menu screen follows:

```
Configure System
    ->System Daemons Configuration
        ->System Daemons Table
```

```
                      System Daemons Table

    TCP   snmpd     YES  *         /etc/snmpd                >
    TCP   -         YES  -         /usr/bin/domainname  ""   >
    TCP   portmap   YES  *         /etc/portmap              >
    TCP   keyserv   NO   *         /etc/keyserv              >
    TCP   ntpd      NO   *         /etc/ntpd                 >
    NFS   nfsd      YES  *         /etc/nfsd            4    >
    NFS   exportfs  NO   *         /etc/exportfs        -av  >
    NFS   mountd    YES  *         /etc/mountd               >
E-> NFS   biod      YES  *         /etc/biod            4    >
    NFS   pcnfsd    NO   *         /etc/pcnfsd               >
    .
    .
    .
```

If you are not using the menu system, edit the /etc/config/daemons file and set the NFS biod daemon to be YES. (Editing this file will ensure that the biod daemon will be started automatically each time you start the UNICOS system in the future.) Then execute the /etc/sdaemon script to start biod now, as follows:

```
# /etc/sdaemon -s biod
```

Note: You cannot do the remaining steps to this procedure by using the menu system.

3. If you do not want to configure UNICOS NFS ID mapping at this time, you should disable this feature by executing the following command (for information on UNICOS NFS ID mapping, see the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304:

```
# /etc/uidmaps/nfsidmap -d
NFS ID mapping is disabled.
```

Note: To disable NFS ID mapping permanently, place the `/etc/uidmaps/nfsidmap -d` command in the `/etc/uidmaps/Set.domains` file; otherwise, if you want NFS ID mapping disabled, you must execute step each time you start UNICOS NFS.

4. Create mount points (empty directories in which the NFS file systems will be accessed on your system) for the file systems you will mount using NFS.

   Example:

   ```
   # cd /UTNA
   # mkdir user1
   # mkdir sunbin
   ```

5. If you would like the file systems to be mounted using NFS automatically when you start the UNICOS system, create the `/etc/mountnfs` script and include the appropriate `mount` commands. Based on the preceding examples, a sample script follows:

   ```
   # cat /etc/mountnfs
   # Script for mounting NFS file systems
   #
   mount /UTNA/user1 &
   mount /UTNA/sunbin &
   ```

6. Ensure that the `/etc/mountnfs` script is executable by executing the following command:

   ```
   # chmod +x /etc/mountnfs
   ```

7. Run the `/etc/mountnfs` script to mount the file systems by executing the following command; when you start the UNICOS system in the future, the script will be run automatically:

   ```
   # /etc/mountnfs
   [1] 85260
   [2] 85261
   ```

8. Verify that the file systems have been mounted by using the /etc/mount and df commands, as shown in the following examples:

```
# /etc/mount
/ on /dev/dsk/root read/write on Mon Jan 10 08:45:33 1994
/tmp on /dev/dsk/tmp read/write on Mon Jan 10 08:46:11 1994
/usr on /dev/dsk/usr read/write,rw on Mon Jan 10 08:46:12 1994
/home on /dev/dsk/home read/write,rw on Mon Jan 10 08:46:13 1994
/usr/src on /dev/dsk/src read/write,rw on Mon Jan 10 08:46:13 1994
/proc on /proc read/write on Mon Jan 10 08:46:14 1994
/els_src on /dev/dsk/els_src read/write on Mon Jan 10 09:14:58 1994
/UTNA/sunbin on tngmoon:/usr/bin read only,ro,soft,bg on Mon Jan 10 18:13:41 1994
/UTNA/user1 on tngmoon:/home/tngmoon/user1 read only,ro,soft,bg on Mon Jan 10 18:13:41 1994
```

```
# df
/UTNA/user1  (tngmoon:/home/tngmoon/user1):
                                  100955  1K blocks ( 48.2%)
/UTNA/sunbin (tngmoon:/usr/bin ):  65879  1K blocks ( 50.7%)
/els_src     (/dev/dsk/els_src ): 178106  4K blocks ( 59.4%)*   57677 I-nodes
/proc        (/proc            ): 119400  4K blocks ( 95.5%)      412 procs
/usr/src     (/dev/dsk/src     ):  27132  4K blocks ( 18.1%)*   25654 I-nodes
/home        (/dev/dsk/home    ): 671031  4K blocks ( 97.8%)*  169883 I-nodes
/usr         (/dev/dsk/usr     ): 152964  4K blocks ( 59.0%)*   26694 I-nodes
/tmp         (/dev/dsk/tmp     ): 495065  4K blocks ( 98.8%)*   98292 I-nodes
/            (/dev/dsk/root    ):  13417  4K blocks ( 17.9%)*   16169 I-nodes
```

Procedure 41: Configuring a CRAY J90 system as an NFS server

The following steps explain how to configure a CRAY J90 system as an NFS server:

1. Describe the file systems you want to allow other systems to mount using NFS by placing entries in the /etc/exports file. (For a complete list of export options, see the exports(5) man page.) You can use the menu system to place your entries in the /etc/exports file or you can do this manually.

   If you are using the menu system, ensure that the menu system has permission to change the configuration for NFS by verifying the NFS configuration option is set to YES in the Configure System->Configurator Automation Options menu. Also, ensure that the Configure System->Major Software Configuration menu has the Network Information Service (NIS) option set to on; if you must change the Major Software Configuration menu, you must rebuild your kernel.

Then select the Configure System->Network Configuration->NFS Configuration->List of Exported File Systems menu, add your entries, and update the form file. Then activate your changes through the NFS Configuration menu. A sample NFS Exported File Systems Configuration menu screen follows:

```
Configure System
   ->Network Configuration
       ->NFS Configuration
           ->List of Exported File Systems
```

```
  NFS Exported File Systems Configuration
    File System  Clients  ro  rw clients  Anon  UID  Root Clients  Sum  Ker
    -----------  -------  --  ----------  ----  ---  ------------  ---  --- >
E->  /home                 rw              anon  -2   edge
```

If you are not using the menu system, ensure that the RC_NFS= parameter is set to YES in the /etc/config/rcoptions file. Then edit the /etc/exports file to describe the file systems you want to allow other systems to mount using NFS.

2. Look at the list of these file systems by using the cat /etc/exports command, as shown in the following example:

   ```
   # cat /etc/exports
   /nasc   -root=edge:sn1234
   /home   -rw
   /tmp
   /UTNA/goodstuff
   ```

3. Make all file systems described in the /etc/exports file available to NFS client systems by using the menu system or doing it manually.

   If you are using the menu system, select the Configure System->System Daemons Configuration->System Daemons Table menu, set the Start up at boot time? option to YES, and update the form file. Then activate your changes through the System

Daemons Configuration menu. A sample System Daemons Table menu screen follows:

```
Configure System
    ->System Daemons Configuration
        ->System Daemons Table
```

```
                  System Daemons Table

S-> Group                                        NFS
     Name                                         exportfs
     Start up at boot time?                       YES
     Kill action                                  *
     Executable pathname                          /etc/exportfs
     Command-line arguments                       -av
     Additional command-line arguments
     Additional command-line arguments
```

If you are not using the menu system, you can make all file systems described in the /etc/exports file available to NFS client systems by executing the /etc/exportfs -av command, as follows:

```
# /etc/exportfs -av
```

Caution: If you are not using the menu system, you **must** run the /etc/exportfs -av command each time your system is rebooted. You should automate the execution of this command by using the menu system.

4. Enable the NFS server daemons; at a minimum, the TCP/IP portmap daemon and the NFS nfsd and mountd daemons must be started on an NFS server. You can use the menu system to enable the daemons or you can do this manually. The cnfsd daemon is necessary **only** when you have more than one Cray Research system; it is intended for use only between Cray Research systems. For more information, see the exports(5) and cnfsd(8) man pages.

   If you are using the menu system, select the Configure System->System Daemons Configuration->System Daemons

Table menu, set the NFS server daemons to YES, and update the form file. Then activate your changes through the System Daemons Configuration menu. When you activate this change, the daemons will be started automatically each time you start the UNICOS system. A sample System Daemons Table menu screen follows:

```
Configure System
   ->System Daemons Configuration
      ->System Daemons Table
```

```
                     System Daemons Table

TCP   snmpd     YES  -            /usr/bin/domainname     ""   >
TCP   -         YES  -            /usr/bin/domainname     ""   >
TCP   portmap   YES  *            /etc/portmap                 >
TCP   keyserv   NO   *            /etc/keyserv                 >
TCP   ntpd      NO   *            /etc/ntpd                    >
NFS   nfsd      YES  *            /etc/nfsd               4    >
NFS   exportfs  YES  *            /etc/exportfs           -av  >
NFS   cnfsd     NO   *            /etc/cnfsd              4    >
NFS   mountd    YES  *            /etc/mountd                  >
NFS   biod      YES  *            /etc/biod               4    >
NFS   pcnfsd    NO   *            /etc/pcnfsd                  >...
```

If you are not using the menu system, edit the /etc/config/daemons file to enable the NFS server daemons, as shown in the following example. (If you do not use the menu system, editing this file also will ensure that the daemons will be started automatically each time you start the UNICOS system.)

```
# vi /etc/config/daemons
TCP     portmap         YES     *               /etc/portmap
NFS     nfsd            YES     *               /etc/nfsd       4
NFS     cnfsd           NO      *               /etc/cnfsd      4
NFS     -               YES     -               /etc/exportfs   -av
NFS     mountd          YES     *               /etc/mountd
```

Then execute the /etc/sdaemon script as follows to start the NFS server daemons (you should have the TCP/IP portmap daemon already running):

```
# /etc/sdaemon -g NFS
```