(Figure Description) Title page illustration

Novell Native File Access for UNIX

www.novell.com

ADMINISTRATION GUIDE

Novell®

August 30, 2001
Novell Confidential

Legal Notices

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

Novell Native File Access for UNIX
July 2001
123-1234-123

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

Access Manager is a registered trademark of Novell, Inc., in the United States and other countries.
ConsoleOne is a trademark of Novell, Inc.
NDS is a trademark of Novell, Inc.
NetWare is a registered trademark of Novell, Inc., in the United States and other countries.
Novell is a registered trademark of Novell, Inc., in the United States and other countries.
Novell Client is a trademark of Novell, Inc.
Novell Directory Services is a trademark of Novell, Inc.
Transaction Tracking System is a trademark of Novell, Inc.
TTS is a trademark of Novell, Inc.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

1 Understanding

Novell® Native File Access for UNIX* provides NFS Server that lets UNIX workstations access and store files on NetWare servers. It is an implementation of the Network File System (NFS) protocol. The required software components are installed and run only on the NetWare® servers without installing additional software on UNIX workstations. The UNIX users attach to NetWare storage using NFS over the TCP/IP protocol. They can mount the exported network storage and use them as their own file system.

The traditional file system is supported only on NFS version 2. The NSS file system, however, is supported on both NFS versions 2 and 3. NFS Server provides mount protocol versions 1, 2, and 3 over UDP. The NFS Server supports NFS protocol versions 2 and 3 on UDP and TCP.

Apart from providing file sharing system between UNIX and NetWare platforms, Novell Native File Access for UNIX provides a complete Novell Directory Services® (NDS®) enabled Network Information Services (NIS) with which UNIX and NetWare users can be administered from a single point, namely NDS. NIS maintains its information in NDS and integrates the user information so that the NDS User object also represents the NIS user.

Main Features of Novell Native File Access for UNIX

- NFS Server
- Network Information Service
- UNIX User Management Using NDS
- ConsoleOne-Based Administration
- Cluster Services Support

NFS Server

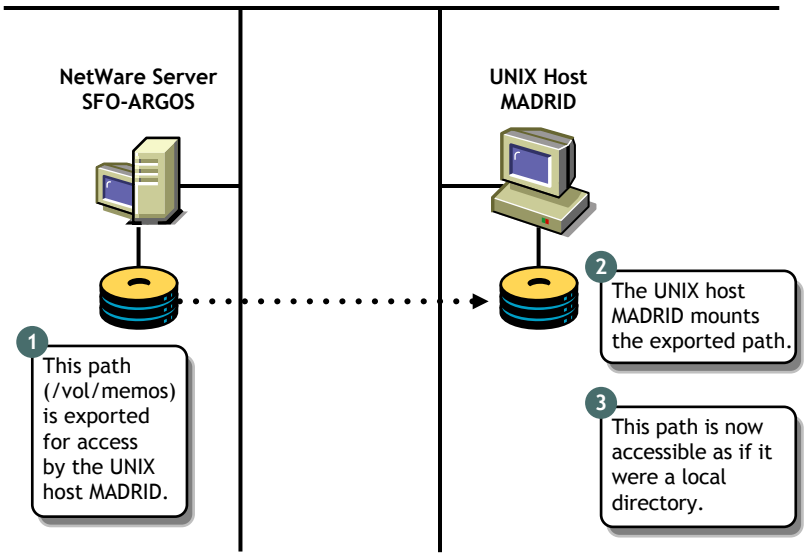
NFS Server software enables UNIX users to access a NetWare file system as if it were a local directory on the UNIX workstation. Any client that supports the NFS protocol can also access NetWare files using the Network File System (NFS) Server.

This section uses the UNIX operating system as the example when referring to the remote NFS client. However, any client that supports the NFS protocol can access NetWare files using the NFS Server.

- NFS Server Operations
- NFS Server Access Control

NFS Server Operations

After exporting the NetWare file system through ConsoleOne, you must mount it to a UNIX workstation. After these processes are completed, UNIX users can manipulate the NetWare files as if they were local UNIX files. The following figure shows an example of the NFS Server file sharing process.



Making NetWare File System Available to NFS Clients

Before UNIX users can access a NetWare file system it must be made available to the UNIX workstations. This process is called *exporting* the file system. When exporting, you can define who should access, and how the information is accessed by specifying the trusted systems and export options. For example, you can restrict the access to specific UNIX workstations, and export the directory as Read-only.

Accessing the NetWare File System from NFS Clients

After exporting a NetWare file system from a NetWare server, you must mount the exported file system on the UNIX workstation for normal access. This process is called *mounting* the file system. Mounting a NetWare file system from a UNIX workstation consists of the following:

- Creating a mount point

  A mount point is an empty directory you create. This directory becomes the access point for the NetWare file system. If you choose an existing directory as a mount point, the contents of the existing directory become unavailable until you unmount the remote file system.

- Mounting the NetWare directory

  Most UNIX systems use the *mount* command to mount a remote file system.

After these steps are complete, UNIX users can access the NetWare file system by accessing the local mount point. Different UNIX systems can use slightly different commands or user interfaces to mount a remote file system.

Web-NFS Access

Web-NFS enables direct Web access to data on NFS servers. It defines a new NFS URL that complements HTTP. Using this URL, browsers with Web-NFS support can access data from any server.

Web-NFS extends NFS to support operations over a WAN. With Web-NFS clients can obtain file handles more easily without going through the portmapper or the mount protocols. This makes it firewall-friendly and enables NFS operations across WANs and the Internet. It also improves performance over a WAN by reducing the number of turnarounds.

For each NFS server, only one of the exported paths can be enabled for Web-NFS access.

NFS Server Access Control

NetWare and UNIX use different methods for controlling access to files. Although both have similar directory and file security, NetWare security is more elaborate. At a basic level, both systems assign access controls to similar user types. The specifics, however, are different. The file sharing service maps these differences so that setting access controls from one system has similar meaning and effect on the other system.

Depending on the level of security and type of access control that suits your network setup, the NetWare NFS file sharing service enables you to select from five different access control mechanisms called *access control modes*.

The following are available access modes:

- NetWare Mode
- NetWare-NFS Mode
- NFS-NetWare Mode
- NFS Mode
- Independent Mode

For traditional volumes all the above modes are available.

IMPORTANT: The Independent Mode is strongly recommended for volumes with large directory entries. Otherwise, performance can suffer due to creation of large number of trustees in the case of NetWare-NFS Mode and NFS-NetWare Mode.

For an NSS volume, only the Independent Mode is supported. In the independent mode, there is no rights/ permissions mapping done. NFS Client rights apply for NFS client access and NetWare rights apply for NetWare client access.

For information about NFS Server Access Control, see Appendix A, “NFS Server Access Modes,” on page 75

For information about NFS Server configuration, and management, see “Setting Up NFS Server” on page 32.

Network Information Service

Network Information Service (NIS) software lets you administer both UNIX and NetWare from a single point, namely Novell Directory Services (NDS).

NIS contains common information about users, groups, and hosts and other information that any client might require. NIS maintains its information in

NDS and also integrates the user information so that the NDS User object also represents the NIS user.In the NDS enabled NIS, all NIS related information is stored as NDS objects. The NetWare NIS can also be set up to work in the various NIS configurations available.

.

NIS Information on NDS

NIS Domain

The NIS system organizes nodes into administrative segments called *domains*. The NIS domain exists only in the local environment and usually covers a single network. An NIS domain is a hierarchical structure, hence it is stored as a container on NDS. NIS does not impose any strict rules on domain naming; however, each domain must have a unique name.

An administrative NIS domain could be a company or a division of a company. Many administrators using DNS choose to relate their NIS domain name to their DNS domain name, but this is not necessary.

NIS Maps

NIS stores all the common information pertaining to a domain as a set of NIS Maps. Users can access the information present in these NIS maps. In the NDS enabled NIS these maps are stored as containers under the NIS domain container. A migration utility is available to create the NIS maps under a specified domain. The NIS Server supports both standard and custom maps.

Standard NIS Maps: Standard maps are created from the standard NIS text files.

The following are the standard maps supported. They are classified according to the type of records they contain:

Ethers Map—A source of information about the Ethernet addresses (48-bit) of hosts on the Internet. The Ether objects (ieee802Device) store information about the Ethernet address and hostname.

Bootparams Map—A source of information for various boot parameters. The Boot objects store information about the boot parameters of the various devices that are running. If the Bootparams text filename is to be migrated from the ConsoleOne, it should be named *bootp*.

Hosts Map—Contains one entry for each IP address of each host. If a host has more than one IP address, it will have one entry for each. The Hosts objects store the IP address and hostname as distinguished values of CN, and aliases and nicknames are stored as other values of CN attributes.

Netgroup Map—A source of information about Net Group parameters. It provides the abstraction of net groups.

Networks Map—Contains a single object for each network. The Network objects store network names as distinguished values of CN, and aliases and nicknames are stored as other values of CN attributes.

Protocols Map—Contains one object for each protocol. The Protocols objects store protocol names as distinguished values of CN, and aliases and nicknames are stored as other values of CN attributes.

RPC Map—Contains one object for each Remote Procedure Call (RPC) program name. The RPC objects store RPC program names as distinguished values of CN, and aliases and nicknames are stored as other values of CN attributes.

Services Map—Contains an object for each service. The Services objects store service names, ports, and protocols as distinguished values of CN, and aliases and nicknames are stored as other values of CN attributes.

Passwd Map—Maintains the details of the users such as UID, User name, home directory etc.

Group Map—Maintains the details of the groups present such as GID, Group name, and Group members.

Ypservers Map—Maintains a list of NIS slave servers for the domain which can also serve the NIS domain.

Custom NIS Maps: You can use NIS to store any common configuration information that is valuable to NIS clients. Maps you create in addition to the standard NIS maps are called *custom maps*. For example, you can create an NIS map that provides employee phonelist.

You can create custom maps by creating a text file that contains the relevant configuration information. After creating the text file, you convert it into an NIS map through migration.

To create a phonelist map, you would begin by creating a text file containing each employee's name and phone number. An NIS map text file must conform to the following rules:

- Each data line begins a new entry key.
- The backslash character (\) at the end of a line appends the next line to the current line.
- The pound sign (#) at the beginning of a line tells the converter to ignore the line.
- Blanks separate the key and the value. Therefore, you must use underscores to replace all other blanks within the key, such as the space between an employee's first and last names. Blanks are acceptable within the key values such as the phonelist.

The following is an example of the phonelist text file:

```
# This is the text file for the phonelist map.

Janice_SmithMS 881-1456

Bob_SpillerMS 235-6777

Jim_Miller MS 769-8909
```

Various NIS Configurations

NIS can be configured in the following ways:

- NIS Master Server
- NIS Slave Server
- NIS Client

NIS Master Server

The master server is the true single owner of map data. It is responsible for all map maintenance and distribution to slave servers. Once an NIS map is built on the master the new map file is distributed to all slave servers for that domain, through client-server relationship. The administrator, must therefore make all the modifications only on the master. The master maintains a list of slave servers, within its domain in the form of a map named Ypservers.

NIS Slave Server

The administrator can set up read-only copies of the NIS database on secondary servers. The secondary servers are referred to as *slaves*. When the server is set up as NIS slave, it contacts the master NIS server and requests a complete copy of the NIS maps on that server.

Once the slave server is set up, you don't need to manage the update process manually. The slave servers periodically query the master and request an update when the slave detects a more recent time stamp on the master.

A slave server can be added to the ypservers map in the master.

It is recommended that you set up at least one slave server for each NIS domain. The slave server can then function as a standby if the master server goes down, although it might not be necessary in all networks. Slave servers can also be used for load distribution in the network. A master NIS server for one domain can also function as a slave NIS server for another domain.

NIS Client

NIS client enables the user to query NIS map information from NIS servers.

For more information on setting up and managing NIS, see "Setting Up NIS Server" on page 45.

UNIX User Management Using NDS

Earlier implementations of NIS on NetWare used to have their own data stores to place the network information.The implementation required two users / groups; one UNIX user /group and one NDS User/group, and these users/ group had to be mapped inorder to represent a single network user / group.

With the current implementation of NIS over NDS, there exists only one user in the network which contains NDS information and UNIX information (stored as the UNIX profile of the user). This brings up the user management to single point, namely NDS.

For this purpose, the NDS schema has been extended and the relevant user information is placed in the NDS Library. The User object now stores UNIX information such as UID, GID, password, home directory, and shell on NDS.

By default, UNIX users /groups are looked for within the default bindery or servers context, but the search path can also be manually configured. Configuration is done using a parameter in the configuration file NFS.CFG, named SEARCH_ROOT. The search is recursive within the containers specified by this parameter.

When a set of users /groups are migrated to NDS from a UNIX server, corresponding User /group objects are created /updated on NDS. During migration, if the new UNIX user or group is not present, a new NDS User or

Group object is created with default NetWare rights. If the User or Group object exists, the user or group's UNIX related information is updated by default during migration.

User and Group Information

NetWare and NFS both use the same User and Group objects to get the information they need.

When a user makes a request to access one of the services, it searches for the User object on NDS by default. The same applies to users who access services as members of a group.

The services can also be configured to look for users, and groups from a remote NIS database.

Information about NFS Users and Groups

The user information includes the following:

- Username
- UNIX User Identification Number (UID)
- Home directory
- Preferred shell
- UNIX Group Identification Number (GID)
- Comments

The Group Information includes the following:

- Groupname
- Group Identification Number (GID)
- Users present in this group

A typical UNIX system stores user account information in the /ETC/PASSWD file and stores group information in the /etc/group file. You can migrate this data directly into NDS using the migration utility.

NFS Usernames, Groupnames, and ID Numbers

Each user uses a username to log in to the system. The UID identifies file and directory ownership information. The user's UID can be a number between 0

and 65,535, with the numbers 0 through 99 usually reserved. (0 is usually assigned to the Superuser.)

NFS groupnames also have identification numbers. The range of numbers is between 0 and 65,535, with the numbers 0 through 99 reserved. The GID identifies the user as a member of the primary group identified by that GID.

User Home Directories

The home directory is the absolute pathname of the user's home directory on UNIX machines.

User Preferred Shells

The shell information identifies the path of the shell program that runs when the UNIX user logs in to the system. You can set the login account to run any program when a user logs in to the system, but the program typically creates an operating system working environment.

Handling UNIX User Password

The current implementation does not migrate the existing UNIX side password field in the password map.

Before migrating the users and groups, remove the password field ("*", or "x") from the corresponding text file and then migrate. After doing this, you can set the UNIX side password. This is done by making the UNIX machine as NIS client to the NetWare machine, logging in as that NIS user and running an NIS client utlity named yppaswd to set the UNIX password.

For information about UNIX user management, see “UNIX User Management” on page 26.

ConsoleOne-Based Administration

Novell ConsoleOne is a Java-based management framework into which various products can plug-in their administrative utility by providing their own snap-ins.

By using ConsoleOne’s snap-in utility for Native File Access for UNIX, you can perform the following tasks:

- Configure the server's global parameters

- Start and stop services
- Configure and manage services
- Configure error reporting
- Monitor performance and adjust parameters affecting performance
- Configure user and group UNIX information

For information about how the ConsoleOne snap-in is used in Native File Access for UNIX, see "ConsoleOne Based Configuration" on page 19.

Cluster Services Support

On a non -cluster environment, if the server running Native File Access for UNIX fails due to some reason, then the UNIX users will not be able to use this service till the NetWare server is up. To achieve high availability we can run Native File Access for UNIX on Novell Cluster Services™.

The product is installed on all the required nodes in the cluster. Cluster enabling is achieved by storing the required configuration files on the shared disk in the cluster. Native File Access for UNIX then access these files through an always or highly available virtual IP address. NFS / NIS clients, must therefore, use the virtual IP Address for NFS mounts / issuing NIS client calls. In case the server where the services are currently running fails, the shared disk volume with configuration files automatically remounts along with the virtual IP on a designated node in the cluster.

Native File Access for UNIX supports only active-passive mode on the cluster. This means that the source and failover destination node cannot be running NFS Services at the same time.

Running Novell Native File Access for UNIX on Novell Cluster Services provides the following benefits:

- There is no need to replicate configuration information as the configuration files are stored on the shared disk.
- Services can be automatically restarted without user intervention in case of a node failure in cluster
- The services can be migrated and controlled between the various nodes in the cluster using ConsoleOne.
- Since the cluster volume is the same regardless of which server it is mounted on, no configuration information is lost or out of date.

For information on configuring Native File Access for UNIX on Novell Cluster Services see “Setting Up on Novell Cluster Services” on page 61

2 Setting Up and Managing

This chapter provides information on setting up, and managing Novell® Native File Access for UNIX*. The following sections explain how to set up and manage Native File Access for UNIX:

- Configuration Methods
- General Parameters
- UNIX User Management
- Setting Up NFS Server
- Setting Up NIS Server

Configuration Methods

Novell Native File Access for UNIX can be configured through ConsoleOne™ and also by setting the file-based configuration parameters of the various components.

ConsoleOne Based Configuration

The ConsoleOne administration utility is a Java* application that allows you to administer various NetWare® products, including Novell Native File Access for UNIX, as well as network resources. It runs on both NetWare servers and clients. Anyone with a NetWare account and access to the server console can run ConsoleOne. To start ConsoleOne from the client, do the following:

IMPORTANT: Before starting ConsoleOne, ensure that you run NFSSTART on the server that you want to administer.

1. Start ConsoleOne from the server where Native File Access for UNIX is installed.
2. Click NFSAdmin and then the login toolbar icon.
3. Enter the tree name, context name, authorized username, and authorized password.
4. Click OK.
5. Enter the hostname or IP address and then click OK.

   IMPORTANT: To login successfully, make sure that your fileserver name and hostname are the same.

File Based Configuration

The configuration (.CFG) files are used to configure the services. All of these files have the following format:

```
PARAMETER_NAME = VALUE
```

Within the .CFG files, a pound sign (#) indicates a comment.

Other than these configuration files, there are specific files for exported volumes for the NFS Server, and for migration utility. All the configuration files are usually located in the SYS:\ETC directory. To configure the modules, you need to change the desired parameter value in the corresponding.CFG file, and restart the module.

General Parameters

The general parameters required by Native File Access for UNIX are located in the NFS.CFG file. These parameters are common to NFS and NIS.If this file is modified, make sure you stop the services using *nfsstop* and restart using *nfsstart*.

File Based Configuration of General Parameters

The following table lists the configuration parameters in NFS.CFG.

Table 1 Novell Native File Access for UNIX General Parameters

Parameter	Default Value	Description
NDS_ACCESS	1	Sets the default access to NDS and all the information is retrieved from NDS. The default value for access to NDS is 1. Set this parameter to 0 and NIS_CLIENT_ACCESS to 1 to get the information from the NIS server.
NIS_CLIENT_ACCESS	0	By default the NIS client access is disabled. Set this parameter to 1 to enable NIS client access.
NIS_DOMAIN		Sets the NIS domain for NIS client access. No default can be provided.
NIS_SERVER		Provides the NIS Server servicing the domain. If a specific server is needed for the domain, this parameter must be set. Otherwise, the NIS server is discovered using the broadcast. No default can be provided.
LOG_FILE_PATH	SYS:ETC\NIS	The path in the NetWare server where you want to write the log file for migration.
MAX_LOG_MSG	5000	Upper limit of number of log messages that can be logged. The information is specific to each log file. By default the last 5000 messages are displayed. If the number of log messages is set to *n*, the last *n* messages are retained.
NIS_LOG_LEVEL	7	The log level indicates the types of messages to be logged. You can either choose one of these or a combination of these. To get the combination, add two or more log levels. For example, to get Error and Information Messages set the Log level to, 5= (1+4). By default, you will get all the messages.

Parameter	Default Value	Description
SEARCH_ROOT		Contains a list of fully distinguished names of containers separated by commas. These containers indicate where the search for users and groups should start. The NDSILIB module uses this parameter. The maximum number of containers allowed is 10 and the string should not exceed 1024 bytes. If you do not set any search containers, search will start from the server's default context.
AUDIT_NETM_LEVEL	0	Specifies the level of alert reported to the SNMP management station. This parameter can take any one of the following values: ◆ 0 (None): Sends no SNMP alerts. ◆ 1 (Critical): Sends fatal SNMP alerts that need urgent attention. ◆ 2 (Major): Sends alerts that must be taken care of immediately. ◆ 3 (Minor): Sends alerts that will not effect operation, but that might degrade performance. ◆ 4 (Informational): Sends alerts that are informational.

ConsoleOne Based Configuration of General Parameters

In this section, the following are explained:

- Viewing the General Server Parameters
- Configuring the General Parameters

Viewing the General Server Parameters

1 In the ConsoleOne main menu, right-click the server you want to configure and then click Properties.

The following panel appears.

These are the general parameters. These fields are read-only.

Host Name—The IP name of the NetWare server.

IP Address—The primary IP address of the NetWare server.

Subnet Mask—The subnet mask that, when added to the IP address, provides the IP network number.

Server Name—The name of the file server.

Operating System—The version of the operating system being used by the host.

Context—The context or logical position of the server within the NDS® tree.

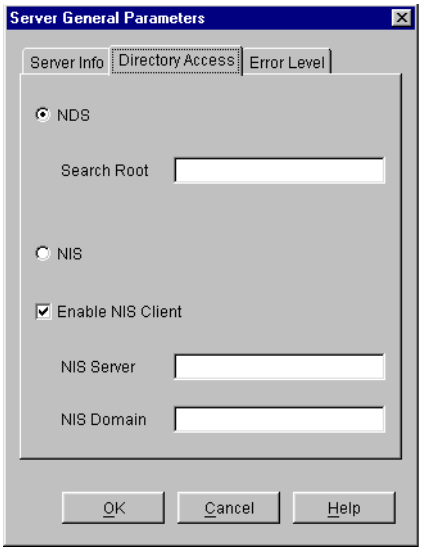
Tree—The current NDS tree.

Time Zone—The world time zone reference for your area. The time zone is used for time stamps and to set time synchronization. The time zone reference is set during installation of NetWare.

Configuring the General Parameters

1 In the ConsoleOne main menu, right-click the server you want to configure and then click Properties > Directory Access.

The following panel appears:



2 This form contains the parameters which can be configured to set the directory access of NetWare NFS Server.

Modify the following Directory Access parameters as necessary:

NDS—Sets the access to NDS.

Search Root—Lists the Fully Distinguished Name of containers from where the search should start for users and groups only. The names are separated by commas. You need to make sure that the parameter has valid values whenever the NDS structure changes.

NIS—Enables remote NIS.

Enable NIS Client—Specifies whether the NIS Client is enabled or not.

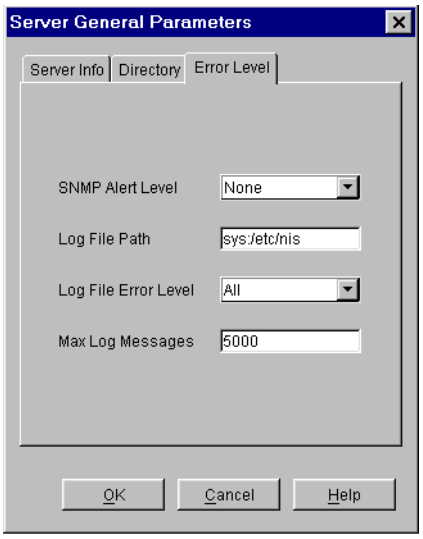
NIS Server—Specifies the remote NIS server name.

NIS Domain—Specifies the domain served by that remote NIS.

3 Click OK.

4 To further specify the server parameters, click Error Level.

The following panel appears:



5 Modify the following parameters as necessary:

SNMP Alert Level—The level of SNMP alerts reported to SNMP management stations. Select an alert level from the drop-down list. You can also turn off SNMP reporting from this list.

- None—Suppresses SNMP reporting.
- Critical—Warns you about urgent problems that require immediate action to prevent widespread failure.
- Major—Warns you about serious problems that require prompt action to prevent failure of the object and possibly some related objects.
- Minor—Provides information about problems that can be addressed as work schedules permit.
- Informational—Provides descriptive information that can be used for such things as trend analysis and planning.

Each level incorporates the information from the levels listed above it. For example, if you select Minor, you also receive messages about major and critical alerts.

Log File Path—The path in the NetWare server where you want to write the log file for migration.

Log File Error Level—The level of error messages written to the AUDIT.LOG file. Select an error level from the drop-down list.

Maximum Log Messages—The maximum number of log messages that can be logged. The information is specific to each log file. By default the last 5000 messages are displayed. If the number of log messages is set to *n*, the last *n* messages are retained.

UNIX User Management

If you are already having an UNIX NIS Server (text based) and you want the new NIS Server to serve the same data served by the old nis server, migration utility helps you.You can copy all those text files into the specified location and run the utility to create NDS entries for a specified domain.

The migration utility creates the Domain object in the default context as well as in two other containers in the same context with the name *domainname*_U and *domainname*_G. While migrating the utility searches for existing NDS users and groups under the containers specified by the SEARCH_ROOT configuration parameter (specified in NFS.CFG) and updates the UNIX information of these objects if they are found. If the objects are not found, the users are migrated to *domainname*_U and the groups are migrated to *domainname*_G. The rest of the data is migrated under the Map objects created under the Domain object.

Maps can be migrated using the following three options:

UPDATE—(Default) Modifies the previous objects' information with the new information, if the object exists, or else, it creates new objects.

REPLACE—Deletes all the previous objects, and new objects are created

MERGE—If previous object exists, retains the old information and creates new objects if old object not found.

Before migrating the users and groups, remove the password field ("*", or "x") from the corresponding text file and then migrate. After doing this, you can set the UNIX side password. This is done by making the UNIX machine as NIS client to the NetWare machine, logging in as that NIS user and running an NIS client utility named yppaswd to set the UNIX password.

For more information on UNIX User Management, see "UNIX User Management Using NDS" on page 14.

File Based Management

Migrating by default uses the makefile SYS:ETC/NIS/NISMAKE, which contains the location of text file for every map. The general usage of migration utility is:

makenis [-[r]d *domainname* [-n *context*] [-f *nismakefilename*] [*mapname* ...]

NOTE: All options should be used only in the specified order.

- In general, to create a domain and migrate data or to use the existing domain object, use the following format:

  `makenis -d domainname`

  The parameter *domainname* is mandatory.

- To remove the existing domain data and then migrate, use the following format:

  `makenis -rd domainname`

- To specify the context where you want to create your Domain object and data, enter it as the contextname:

  `makenis -d domainname -x contextname`

  In the context name prefix each of the dots in the Relative Distinguished Names with a backslash (\) to distinguish them from NDS names.

- To specify an NIS makefile other than the default SYS:ETC/NIS/NISMAKE, use the following format:

  `makenis -d domainname -f makefilepath`

  To specify the text files that you want to migrate, modify the NIS makefile. The NIS makefile is in the following format:

  ```
  map name    full path    parameters (if any)
  ```

  The comment character is the pound sign (#).

  If nothing is specified, all the files in the makefile are migrated.

  For each map, you can specify the SECURE parameter so that only requests coming from secure ports are able to access the data.

  For the Password map, you can specify two additional parameters: -u *uid* (which stops users with a UID less than a particular value from migrating to NDS) and AUTOGEN (which generates a UID from the program itself).

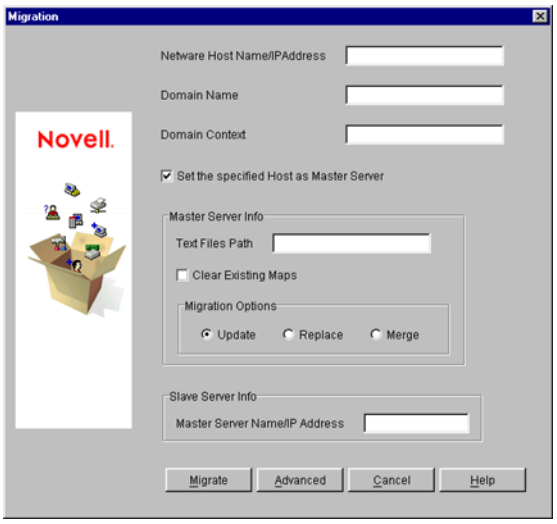
You must specify the text file in the full path in DOS name format.

- To migrate specific maps, use the following format:

  `makenis -d domainname mapname1, mapname2`

ConsoleOne Based Management

1. In the left panel of the ConsoleOne, click The Network.
2. Select the servers' tree where you want to manage the domains and maps.
3. Click the toolbar M icon. The following panel appears.



4. To migrate a domain, enter the NetWare Host Name/IP Address, Domain Name, and Domain Context.
5. To set the NIS Server as master for this specified domain, check Set the Specified Host As Master Server.
6. Enter the Text Files Path in the Master Server Info section.
7. If you want to clear the maps already present, check Clear Existing Maps.
8. Click the radio button for the type of the migration you want to do.
9. To set the NIS Server as Slave Server, enter the name of the Master Server Name/IP Address in the Slave Server Info section.
10. To migrate the domain for default maps, click Migrate.

The available default maps are ethers, hosts, networks, protocols, RPC, services, passwd, group, netgroup, and bootparams. By default these files should be present in SYS:\ETC\NIS.

11 To migrate the domain for specific maps, click Advanced to go to the Map Information panel.

11a Click either Default Maps or Other Maps.

11b Select the desired maps from the list, deselect the maps you do not want to migrate, and click OK.

12 To modify an existing map or add a new map, click Add to go to the Add Map panel.

12a Enter the Map Name and the Absolute Text File Path with the file name.

12b If you want to secure the map, click Secure.

12c In the Comment Character box, enter the comment character present in the specified text file and click OK.

The default character is #.

13 Click Migrate.

Managing Users and Groups

You can add and modify the information of a User or Group object that already exists in NDS.

Modifying User Information

1 In the left panel of the ConsoleOne main menu, click the NDS tree where the object resides.

If you do not find the tree, click Novell Directory Services and then select the tree and log in to it.

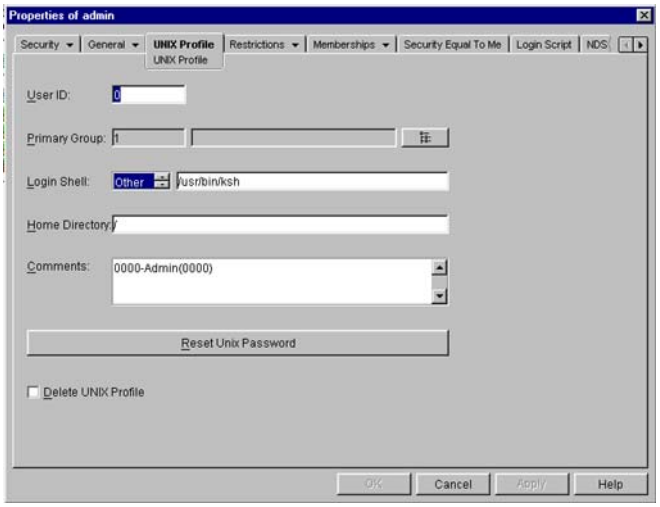
2 Double-click the container named *domainname*_U, where the User objects reside.

The User objects under this particular container appear.

3 Right-click the User object whose properties you want to change and click Properties.

The following panel appears, showing the various forms that should be specified to add and modify the user information on NDS.

All the forms except the UNIX Profile form are standard forms.



4 To modify the UNIX user profile, click UNIX Profile and specify the information in the following fields:

User ID—The users' UNIX UID

Primary Group—The group ID (GID) of the group this user belongs to. To enter the GID of the user, click Browse and select the appropriate group.

Login Shell—The preferred login shell of the user.

Home Directory—The home directory the user wants to be placed in while logging in to the system.

Comments—Any other comments that the user might want to specify.

Reset UNIX Password—Use to reset the user's UNIX password.

5 Click Apply > OK.

Modifying Group Information

1 In the left panel of the ConsoleOne main menu, click the NDS tree where the object resides.

If you do not find the tree, click Novell Directory Services and then select the tree and log in to it.

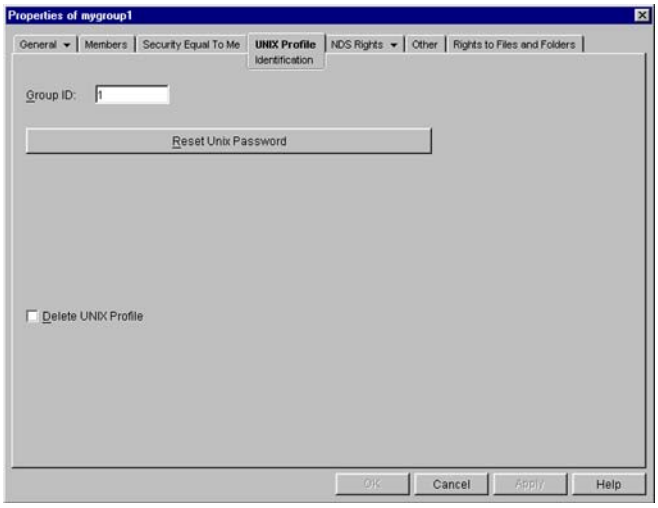
2 Double-click the container *domainname*_G, where the Group objects reside.

The groups under this particular container appear.

3 Right-click the Group object whose properties you want to change and click Properties.

The following panel appears, showing the various forms which should be specified to add and modify the group information in NDS.

All the forms except the UNIX Profile form are standard forms.



4 To modify the UNIX group profile, click the UNIX Profile tab and specify the information in the following field:

Group ID—The groups' UNIX GID.

5 Click Apply > OK.

Adding a New User or Group

To add a new user, do the following:

1. In the left panel of the ConsoleOne main menu, click the context where you want to add the new user.
2. Select File > New, and then click User.
3. Enter the user information.

To add a new group, do the following:

1. In the left panel of the ConsoleOne main menu, click the context where you want to add the new group.
2. Select File > New, and then click Group.
3. Enter the group information.

IMPORTANT: When any update to a UNIX profile is done from ConsoleOne, you must perform NFSSTOP and NFSSTART, which will also allow the new UNIX profile to be seen by the modules.

Setting Up NFS Server

The NFS Server uses the following files:

- NFSSERV.CFG which contains the configuration parameters
- NFSEXPRT which contains the exported path information
- NFSTHOST which contains the trusted hosts list for the exported path

For more information on NFS Server, see "NFS Server" on page 8

File Based Configuration for NFS Server

NFS Server Configuration Parameters

The following table lists the parameters that can be set in NFSSERV.CFG:

Parameter	Default Value	Range	Description
REQ_Q_FULL_ALERT	90	20 - 99	Minimum percentage of request queue utilization which triggers an SNMP alert.
REQ_CACHE_FULL_ ALERT	90	20 - 99	Minimum percentage of request cache utilization which triggers an SNMP alert is sent.

Parameter	Default Value	Range	Description
OPEN_FILE_CACHE_ FULL_ALERT	90	20 - 99	Minimum percentage of open file cache utilization which triggers an SNMP alert is sent.
OPEN_FILE_CACHE_ ENTRIES	512	32 - 1024	Number of open file cache entries.
CACHE_AGING_ INTERVAL	60	0 - 2000	Duration (in seconds) the NFS server keeps a file's information in cache memory. The value 0 disables the open file cache.
REQ_CACHE_ ENTRIES	256	64 - 512	Number of request cache entries.
CACHE_WRITE_ THROUGH	NO	YES / NO	Indicates whether cached data should be written to disk immediately.
TYPE_OF_ TRANSPORT	BOTH	TCP, UDP, or BOTH	Whether the NFS Server should support TCP, UDP, or BOTH.
NFS_VERSION	0	0/2/3 (0 = Both, 2 = only V2, and 3 = only V3)	Indicates which version of NFS protocol should be currently supported.
NFS_UMASK	022	000 - 777	File mode creation mask for default UNIX permissions.
NFS_V2_THREADS	5	1 -150	Number of NFS Server threads servicing the NFS 2 protocol.
NFS_V3_THREADS	5	1 - 150	Number of NFS Server threads servicing the NFS 3 protocol.
MOUNT_V2_THREADS	1	1 - 150	Number of threads servicing Mount V2 requests.
MOUNT_V3_THREADS	1	1 - 150	Number of threads servicing Mount V3 requests.
NFS_V2_TCP_SEND_Q_ ENTRIES	30	1 - 150	Size of the TCP send queue for the NFS V2 protocol.
NFS_V3_TCP_SEND_Q_ ENTRIES	30	1 -150	Size of the TCP send queue for the NFS V3 protocol.

Parameter	Default Value	Range	Description
NFS_V2_RECV_Q_ENTRIES	20	1 - 150	Size of the receive queue for the NFS V2 protocol.
NFS_V3_RECV_Q_ENTRIES	20	1 - 150	Size of the receive queue for the NFS V3 protocol.
LOG_DIR	SYS:\ETC		Directory where the NFS Server creates the log file.
LOG_FILE	NFSSERV		The name of the NFS server log file. A .LOG extension is automatically added to the file.
LOG_LEVEL	7	1 = Error Messages, 2 = Warning Messages, 4 = Information Message	The log level indicates the types of messages to be logged. You can either choose one of these or a combination of these. To get the combination, add two or more log levels. For example, to get Error and Information Messages set the Log level to, 5= (1+4). By default, you will get all the messages.

Exporting NetWare Volumes and Directories

The Export Path information file, NFSEXPRT, contains the list of the paths that are exported from the system. It also gives the specified properties for the exported path.

This file contains one exported path per line. The format of each line is as follows:

`ExportedPath isReadonly anonymousAccess mode webccess`

- **Exported Path**—The directory path to be exported. For example */nfsvol*.
- **isReadOnly**—Specifies whether to export the path in read-only mode or not. Values = 1 (read-only), 0.
- **anonymousAccess**—Specifies whether anonymous access to the exported path is allowed or not. Values = 1, 0.
- **mode**—Specifies the rights and permission mapping modes for the directory. Novell Native File Access for UNIX supports both traditional and NSS file systems. NSS supports only independent mode (value 512),

whereas traditional volumes support other modes as well. The following are values that traditional volumes can take:

- NetWare Mode 12
- Independent Mode 512
- NetWare-NFS Mode 0
- NFS Mode 256
- NFS-NetWare Mode 16

- **Web**—Specifies if Web access is allowed for this exported path. At any point in time, only one path can be enabled for Web access.

Example of an exported path:

```
/nfsvol 0 1 512 0
```

NFS Trusted Host File

The NFSTHOST file contains the list of all the trusted hosts that can access the exported directory. This is specified in conjunction with the nfsexprt file.

The format of every line is as follows:

```
Exported Path Host Name Access-Type Host/Hostgroup
```

- **Exported Path—**Gives the directory path to be exported. For example. / nfsvol.
- **Host Name—**Gives access to the client host named by the user. To give access to all hosts, select (*).
- **Access Type (1, 2, 3)—**Specifies the type of access to be granted to a specific host. The values it can take are as follows:
  - Trusted 1
  - RootAccess 2
  - ReadWriteAccess 3
- **Host/Hostgroup (1, 0)—**This field shows whether the Host Name specified is a Host or a Hostgroup. This field should always be set to 1 (Host).

Example of an exported directory:

```
/nfsvol nfs-sun2 3 1
```

```
/nfsvol nfs-sun2 2 1

/nfsvol nfs-sun2 1 1

/nfsvol * 3 1

/nfsvol * 2 1

/nfsvol * 1 1
```

Removing an Exported Path

To remove an exported path, delete the corresponding directory entries from the files nfsthost, and nfsexprt.

Getting the UNIX information from Remote NIS

For file system sharing by NFS server, the UNIX user, and group information is obtained from NDS by default. This can be modified so that UNIX information is obtained from a remote NIS server. To set this do the following:

1. Do nfsstop
2. In the NFS.CFG file set the parameters as follows:
   - NDS_ACCESS=0
   - NIS_CLIENT_ACCESS=1
   - NIS_DOMAIN= *nis domainname*
   - NIS_SERVER= *servername which is servicing the specified domain*
3. Do nfsstart.
4. Load nfsserv.

ConsoleOne Based Configuration for NFS Server

This section describes how to manage the NFS Server from ConsoleOne.

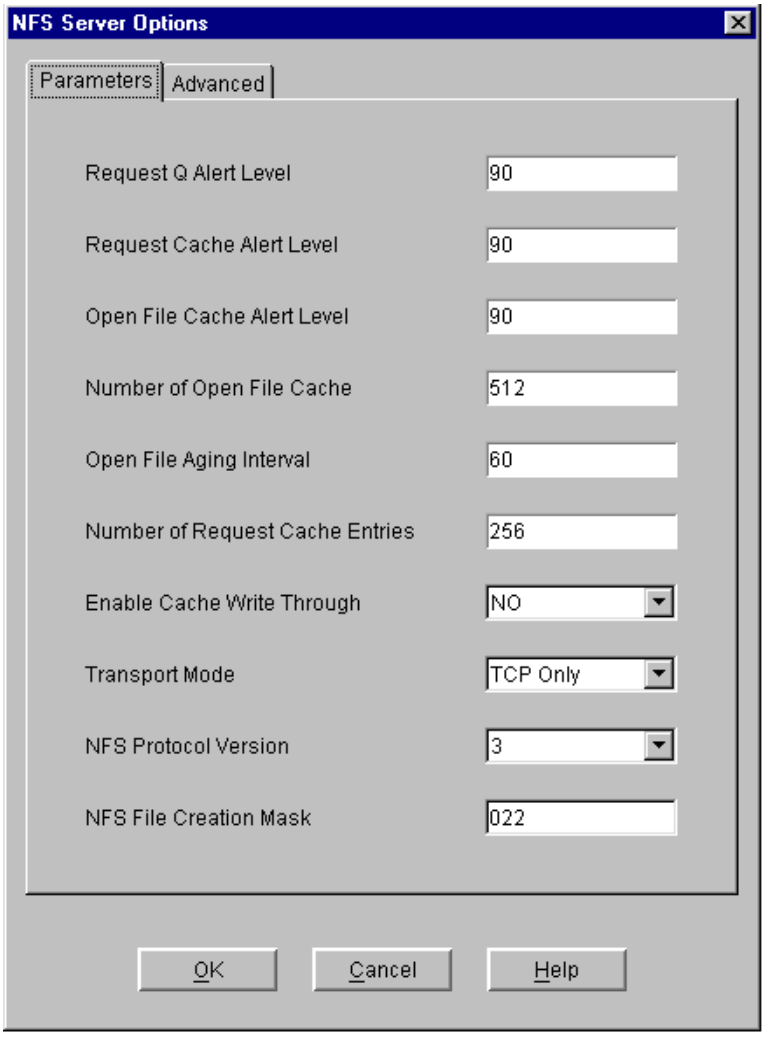
NFS Server General Configuration Parameters

1. After logging in, click the server you want to administer from the list of servers under NFSAdmin in the ConsoleOne left panel.

   The NFS Server toolbar icon and the NFS Server on the menu bar are displayed.

2 To administer NFS Server, click NFS Server on the menu bar and then click Options.

The following panel, which shows the NFS Server basic parameters and their default values, appears.



3 Modify the following parameters as necessary:

Request Q Alert Level—After what percentage of request queue utilization an SNMP alert is sent. Default = 90. Range = 20 - 99.

Request Cache Alert Level—After what percentage of request cache utilization an SNMP alert is sent. Default= 90. Range = 20 - 99.

Open File Cache Alert Level—After what percentage of open file cache utilization an SNMP alert is sent. Default = 90. Range = 20 - 99.

Number of Open File Cache—Number of files the NFS server can have open simultaneously. Default = 512. Range = 32 - 1024.

Open File Aging Interval—How many seconds the NFS server keeps a file's information in cache memory. When a file is held in cache, NetWare users cannot access it. Larger values produce better performance, but they also make NetWare users wait longer to access files that are being manipulated by NFS. Default = 60. Range = 0 - 2000. Open File Caching is disabled at 0.

Number of Request Cache Entries—Number of requests that can be held in cache memory. Default = 256. Range = 64 - 512.

Enable Cache Write Through—Whether cached data should be written to disk immediately or not. By default, the data is not written immediately.

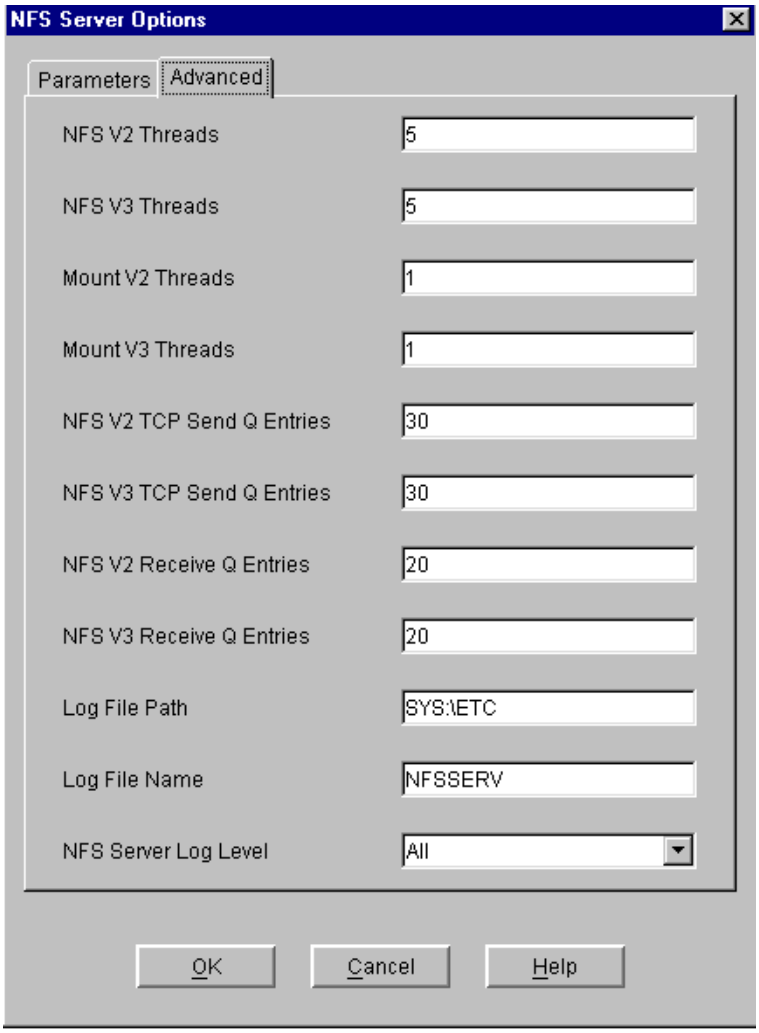
Transport Mode—Which transport mode NFS Server should support. The modes could be UDP, TCP, or Both. Default = Both.

NFS Protocol Version—Version of the NFS protocol to be loaded. The values are 0/2/3.

NFS File Creation Mask—File mode creation mask in Independent Mode for default UNIX permissions of files and directories created from the NetWare side.

4 To specify the advanced parameters, click Advanced on the NFS Server Options panel.

The following panel, which shows the NFS Server advanced parameters and their default values, appears.



5 Modify the following parameters as necessary:

NFS V2 Threads—Number of NFS Server threads servicing the NFS 2 protocol. Default = 5. Range = 1 - 150.

NFS V3 Threads—Number of NFS Server threads servicing the NFS 3 protocol. Default = 5. Range = 1 - 150.

Mount V2 Threads—Number of NFS Server threads servicing the Mount V2 Requests. Default = 1. Range = 1 - 150.

Mount V3 Threads—Number of NFS Server threads servicing the Mount V3 Requests. Default = 1. Range = 1 - 150.

NFS V2 TCP Send Q Entries—Size of the TCP send queue for the NFS 2 protocol. Default = 30. Range - 1 - 150.

NFS V3 TCP Send Q Entries—Size of the TCP send queue for the NFS 3 protocol. Default = 30. Range = 1 - 150.

NFS V2 Q Entries—Size of the receive queue for the NFS 2 protocol. Default = 20. Range = 1 - 150.

NFS V3 Receive Q Entries—Size of the receive queue for the NFS 3 protocol. Default = 20. Range = 1 - 150.

Log File Path—Directory that NFS Server creates the log file in. Default directory is SYS:\ETC.

Log File Name—Name of the NFS Server Log File. Default name is NFSSERV. A .LOG extension is automatically added.

NFS Server Log Level—Indicates the types of messages to be logged.

6 Click OK.

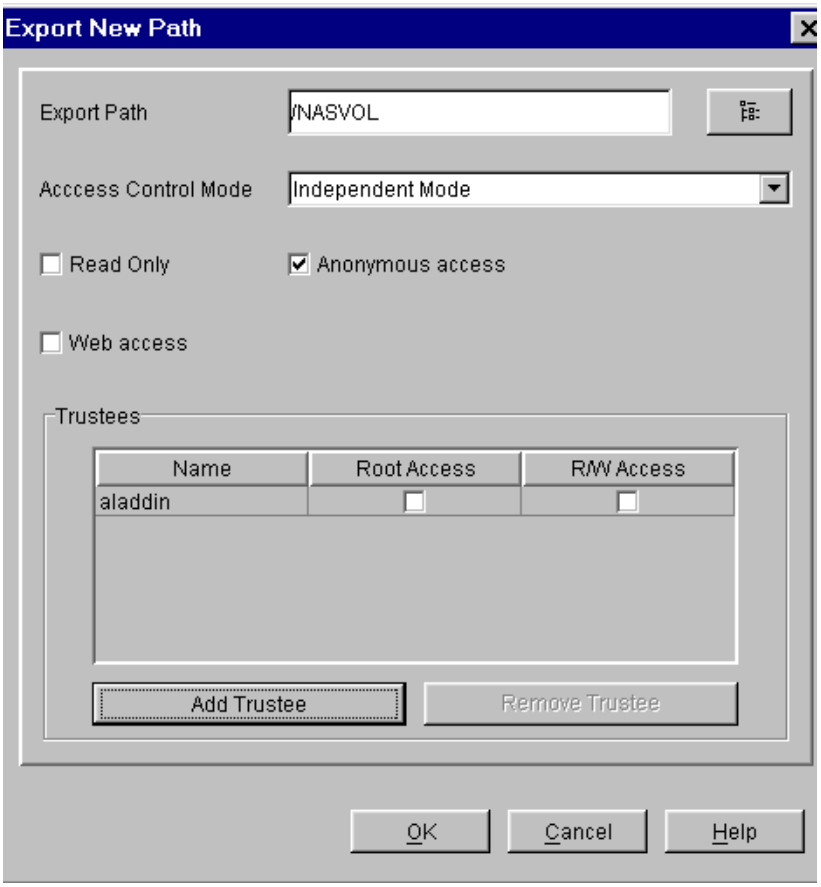
Exporting NetWare Volumes and Directories

Exporting a directory enables NFS client users to view NetWare volumes and directories as part of the client file system.

You can export a NetWare path and manage it.

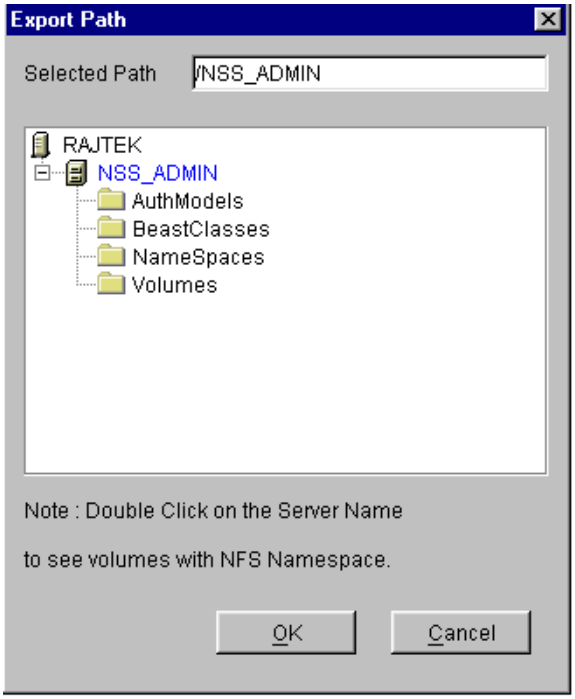
1 Make sure you have added the NFS name space, and then select Export New Path from the NFS Server drop-down list.

The Export New Path panel appears.



2 To export a new directory, click the Browse icon in the upper-right corner of the panel.

The Export Path panel appears.



3 Double-click the server name to see the volumes with NFS name space.

4 Select the volume or directory you want to export and click OK.

5 On the Export New Path panel, modify the following fields as necessary:

Export Path—Path of the directory to be exported.

Access Control Mode—The access control mode that applies to this directory. Select this field to display the possible access control modes (described below). Certain NetWare file attributes can override the file owner's read/write access.

The default is Independent Mode. For NSS volumes only Independent Mode is supported.

- NetWare Mode—Enables you to control access to the exported NFS directory using traditional NetWare access control methods such as NetWare rights and attributes. NFS permissions do not modify the settings of the NetWare attributes. A change in some NetWare attributes can change the UNIX ownership of a directory and make the directory unwritable.

- NetWare-NFS Mode—Creates trustee rights to emulate NFS permissions whenever permissions are created or changed by the NFS client. NFS permissions override NetWare rights, but NetWare attributes are always enforced and, therefore, cannot be overridden by NFS permissions. Select this mode only if you want NetWare trustee rights to emulate NFS permissions.
- NFS-NetWare Mode—Creates trustee rights and NetWare attributes to emulate NFS permissions whenever permissions are created or changed by the NFS client. NFS permissions override both NetWare rights and NetWare attributes. This mode allows both NetWare trustee rights and NetWare attributes to emulate NFS permissions.
- NFS Mode—Does not automatically map between file systems. Select this mode if the exported directory is accessed primarily by NFS clients.
- Independent Mode—This mode is an extension of the NFS Mode. Like the NFS Mode, this mode stops access mapping between NetWare and NFS. It controls NetWare access by creating trustee rights for files independent of their NFS permissions. This mode ensures mapping of the NetWare file owner to the UNIX UID.

Read-Only—Indicates whether user access is limited to read-only. Selecting No (the default) provides all users with read/write access. Selecting Yes limits users to read-only access. If Yes is specified, even users on hosts identified as trusted are limited to read-only access. The same also applies to root users. To override this option, enter the name of that host in the Hosts with Read-Write Access field.

Anonymous Access—Indicates whether the users Nobody and Nogroup can access the exported path. Selecting Yes (the default) provides these users with access. Selecting No denies access.

Web Access—Enables WebNFS access for the selected directory when checked. At any point in time only one of the exported paths can be enabled for Web Access.

6 Click Add Trustee. Enter the hostname that you want to give exported directory/volume access to.

An asterisk (*) will give access to all the hosts.

You can also specify the type of access you want to give to the host.

7 If trustees are already assigned, click the Trustee name on the Export New Path panel to set their access rights. The following panel appears.

Trustees—The hosts which are allowed to access this exported directory. Asterisk (*) indicates that access is given to all hosts.

Hosts with Root Access—The host whose users with root privileges have Admin rights to the exported directory. Select this field to display a list of these hosts. If a host with access is not specified as having root access, root users on that host have the rights of the NFS user Nobody.

Hosts with Read-Write Access—The hosts with access whose users have read/write access to the exported path. Select this field to display a list of these hosts.

8 To remove a host from the Trustee list, select the trustee and click Remove Trustee.

Modifying the Exported Path

1 In the left panel of the ConsoleOne main menu, click the server that you want to administer.

The Export icon appears in the right panel.

2 Double-click Exports to see the currently exported path.

3 Right-click the exported path you want to modify and then click Properties.

You can now see the properties of the exported path and modify them.

4 Make the changes as required and then click OK.

Removing an Exported Path

1 In the left panel of the ConsoleOne main menu, click the server that you want to administer.

The Export icon appears in the right panel.

2 Double-click Exports to see the currently exported path.

3 Right-click the exported path you want to delete and then click Remove.

Getting the UNIX information from Remote NIS

For file system sharing by NFS server, the UNIX user, and group information is obtained from NDS by default. This can be modified so that UNIX information is obtained from a remote NIS server. To set this do the following:

1 Do nfsstop

2. Set the parameters in the NFS.CFG file as follows by following the steps 1 to Step 5 in "Configuring the General Parameters" on page 23
   - NDS_ACCESS=0
   - NIS_CLIENT_ACCESS=1
   - NIS_DOMAIN= *nis domainname*
   - NIS_SERVER= *servername which is servicing the specified domain*
3. Do nfsstart.
4. Load nfsserv.

Setting Up NIS Server

To set up NIS Server you need to first set the general parameters and then, modify the NIS related parameters in NIS.CFG. This file is modified by nisinst.nlm. Load *nisserv* to start the NIS Services. When the parameter values are modified, you need to unload and load *nisserv*.

For information about NIS, see "Network Information Service" on page 10.

File Based Configuration for NIS Server

The following table lists configuration parameters in NIS.CFG.

Table 2 NIS Parameters

Parameter	Default Value	Description
NIS_SERVER_CONTEXT	novell	The NDS context where the NIS server object is created. It holds all the domain FDNs and the NIS server reads the domains from here.
NIS_SERVER_NAME	NISSERV_*ServerName*	The name by which the NIS server will be referenced.
INTERDOMAIN_RESOLUTION	1	Specifies whether interdomain resolution is allowed or not. If allowed, DNS is contacted for hostname resolution even if NIS is not running. This is used for host maps only.

Parameter	Default Value	Description
FILEMARK_LOG_FREQ	100	Puts the file in the log after parsing the specified number of records. This is used by the migration utility when the administrator wants to migrate maps which have large records. After transferring a number of records successfully, an index is maintained. If a transfer breaks, it can start from the index kept previously.
MAP_REFRESH_DEFAULT	24:00:00	Specifies the default time interval for refreshing the maps by synchronizing the maps in the slave server with the master.
NIS_ADMIN_OBJECT_ CONTEXT	novell	The context where the NIS Admin object will be created.

Setting Up a NetWare Server As a NIS Master

1. Create a text file called YPSERV in SYS:\ETC\NIS. For every slave server and enter the hostname of the slave server in this file in the following format:

   slaveserverhostname1 slaveserverhostname1

2. Enter the YPSERVERS map entry in SYS:\ETC\NIS\NISMAKE with its path in the following format:

   YPSERVERS SYS:\ETC\NIS\YPSERV

3. If the domain is already migrated, migrate it again to migrate the YPSERVERS map. For migration, see "File Based Management" on page 27.

   If the domain is not already migrated, copy the NIS related text files such as passwd and group (which are available in /etc in UNIX.) into SYS:\ETC\NIS of NetWare server, and then migrate a domain.

4. Load NISSERV.NLM.

Setting up NetWare Server As NIS Slave Server

1. While setting up the UNIX machine as the master, add the NetWare machine name to the slave server list.

2. In the NetWare machine, enable the NIS_CLIENT_ACCESS by setting the NIS_CLIENT_ACCESS parameter in the SYS:\ETC\NFS.CFG file to 1.
3. Set the domain to one that is being served by the UNIX NIS server, using the following command:

   ypset *domainname hostname*
4. Run NISSERV.NLM.
5. Run MKSLAVE with the following parameters:

   mkslave -d *domainname* -m *master* [-x *contextname*]

NOTE: For the maps Group and Passwd, we do not recommend using the NIS server as a slave. The reason is that in order to serve the NIS client calls of specific domain calls, the slave server looks for the records in all its domains and displays the records of all the domains.

Setting Up As NIS Client

1. Run NFSSTOP.
2. Set NIS_CLIENT_ACCESS=1 in SYS:\ETC\NFS.CFG and uncomment it
3. Run NFSSTART.
4. Set the default domain using ypset by entering

   ypset *domainname hostname/IP_address*

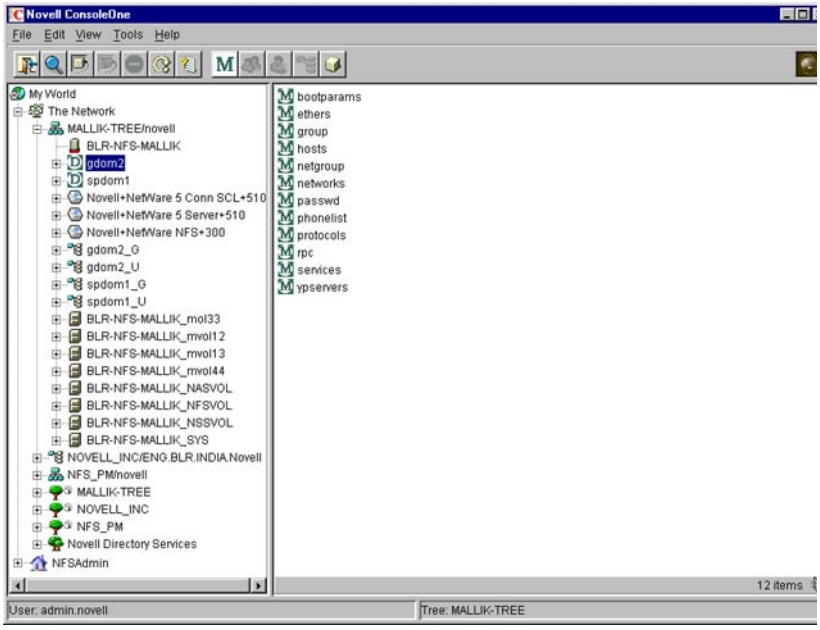
ConsoleOne Based Administration for NIS Server

This section describes how to manage the NIS Server of Novell Native File Access for UNIX as a master/slave server. The section also provides instructions for using an NIS master server on another machine. You should already be familiar with the concept of NIS and understand how information is stored on NDS.

Administering a Domain

1. Complete Step 1 through Step 3 on page 28.
2. Return to the main menu where you can see the migrated domains.
3. Click the domain to see the maps available under it.

   The following panel appears.



4 To see the properties of any map, right-click the map and then click Properties.

5 Modify the following general properties of each Map object as necessary:

Map Master—The name of the master server serving this map.

Map Last Modified—The last time the map was modified by adding or removing records.

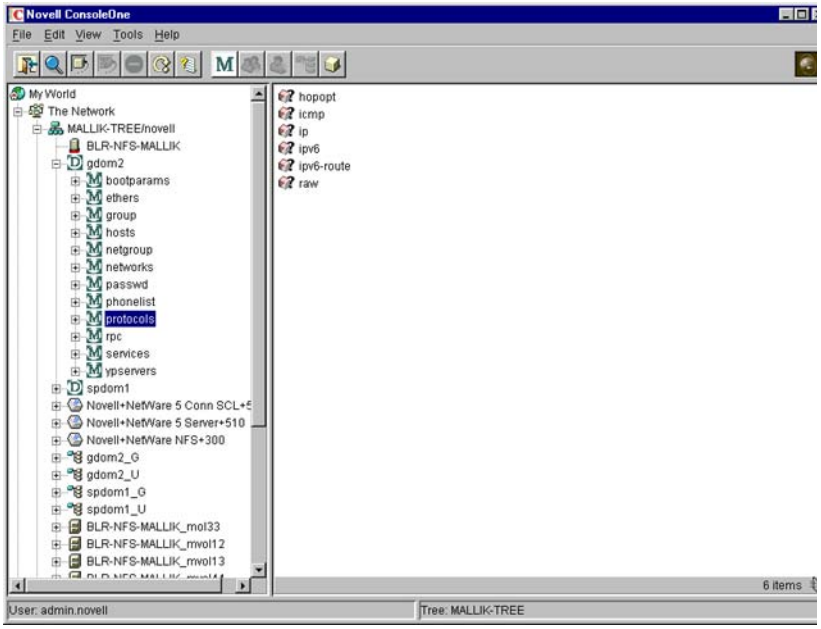
Map Refresh Frequency—The frequency (in hours) at which you want to refresh the maps.

Is Map Secure—Sets the secure flag of the map when checked.

Description—Any general comments that you want to mention.

6 Click each map to perform operations on it and to see the records present under the map.

After you click a map, a panel similar to the following appears:



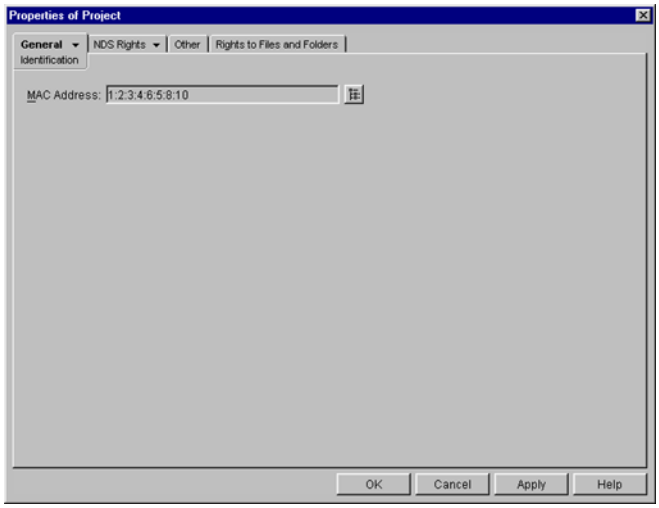
7 To add an object to a map, right-click the map in the left panel, click New, and then specify the details of the object in the dialog box.

While the panels for records on the same map are the same, they differ from map to map.

Administering Maps

The following figures show the main map panels and are followed by procedures for using each panel's basic fields. Using these panel, you can view, or modify the map records' properties. The standard fields remain the same.

Figure 1 Ethers Map Records Properties Panel

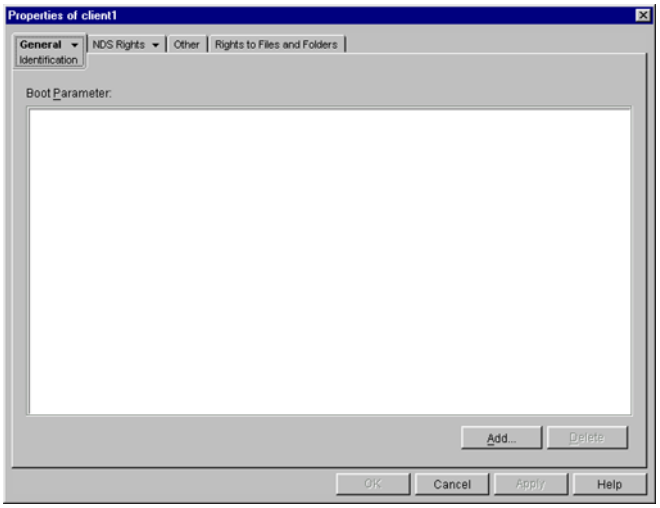


This panel shows the Ethernet address of the host.

The standard address form is *x*:*x*:*x*:*x*:*x*:*x*, where *x* is a hexadecimal number.

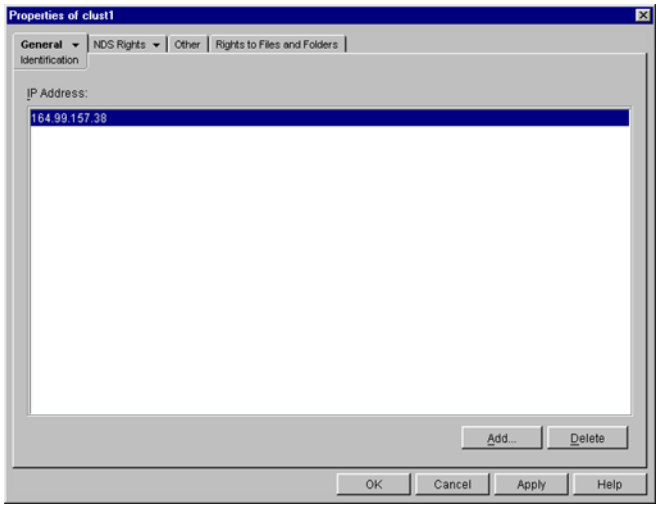
Click the icon to enter the Ethernet address of the host, and then click Apply > OK.

Figure 2 Bootparams Map Records Properties Panel



1. To add the device's boot parameter, click Add, enter the boot parameter of the device in the Boot Parameter field, and then click Apply > OK.
2. To delete the device's boot parameter, select the boot parameter of the device in the Boot Parameter field, and then click Delete > Apply > OK.

Figure 3 Host Map Records Properties Panel

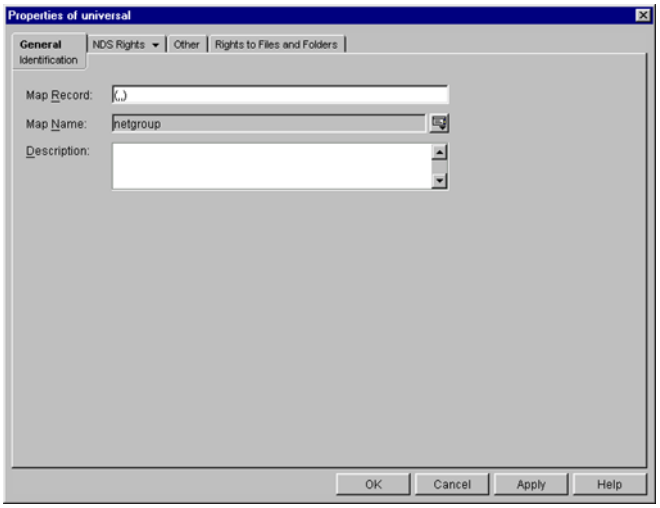


1. To add the host address, click Add, enter the IP address of the host, and then click Apply > OK.

   The network addresses are written in the conventional decimal dot notation.

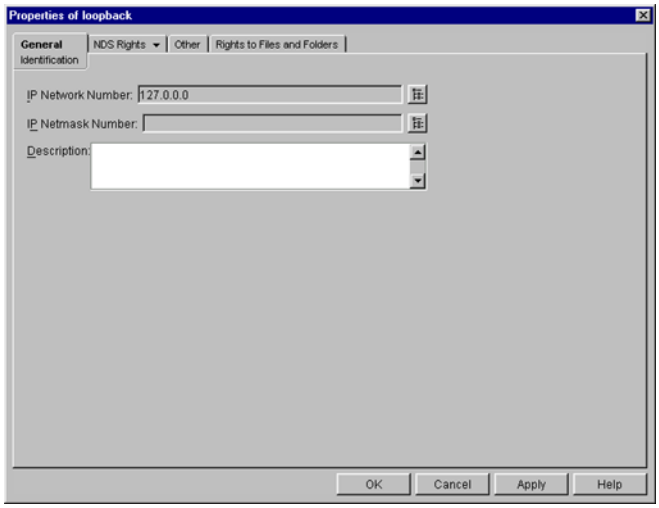
2. To delete the host address, select the host's IP address from the IP Address field, and then click Delete > Apply > OK.

Figure 4 Netgroup Records Properties Panel



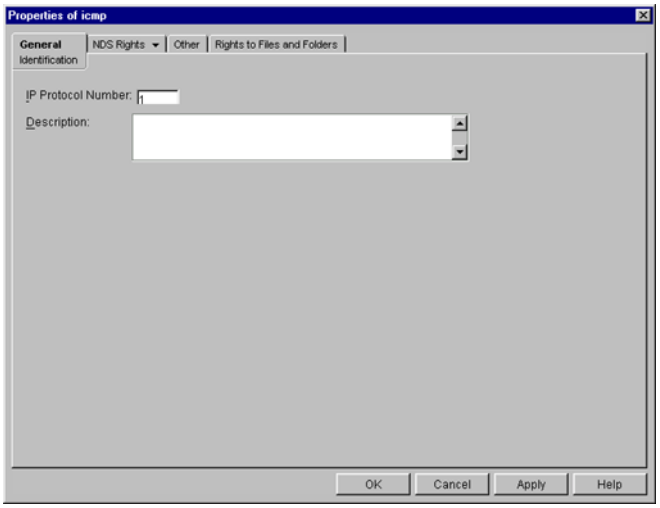
To add a netgroup address, enter the name of the Map Record, browse the icon for the Map Name, enter the description of the map, and then click Apply > OK.

Figure 5 Networks Map Records Properties Panel



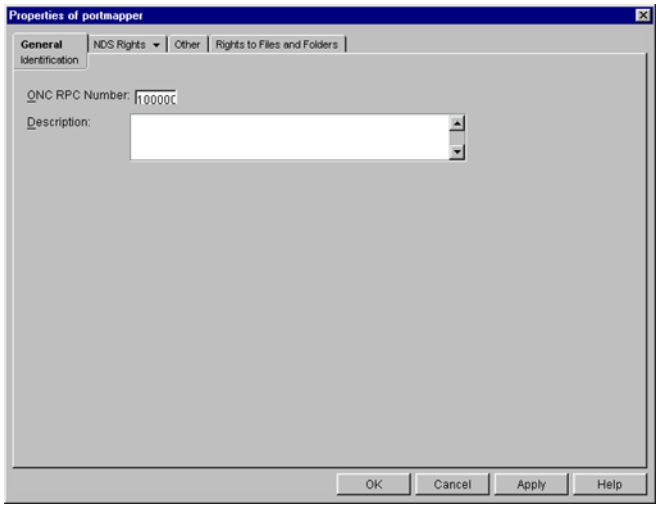
1. To enter the IP network number, click Browse, enter the network number, and click OK.
2. To enter the IP netmask number, click Browse, enter the netmask number, click OK, enter the description of the record, and then click Apply > OK.

Figure 6 Protocols Map Records Properties Panel



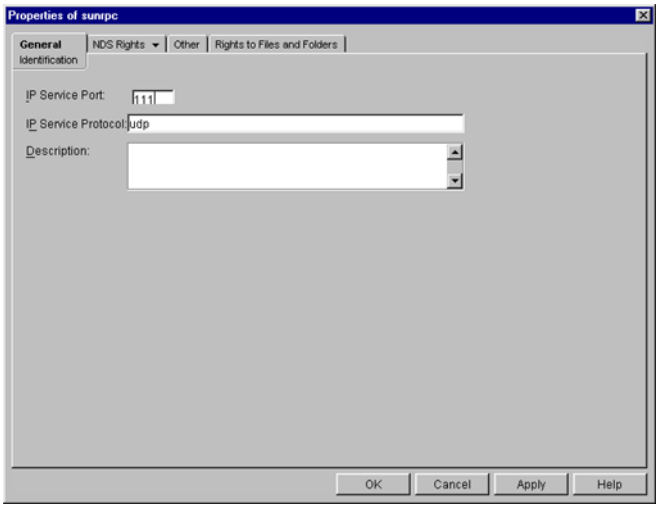
1. Enter the protocol number and a brief description of the record.
2. Click Apply > OK.

Figure 7 RPC Records Properties Panel



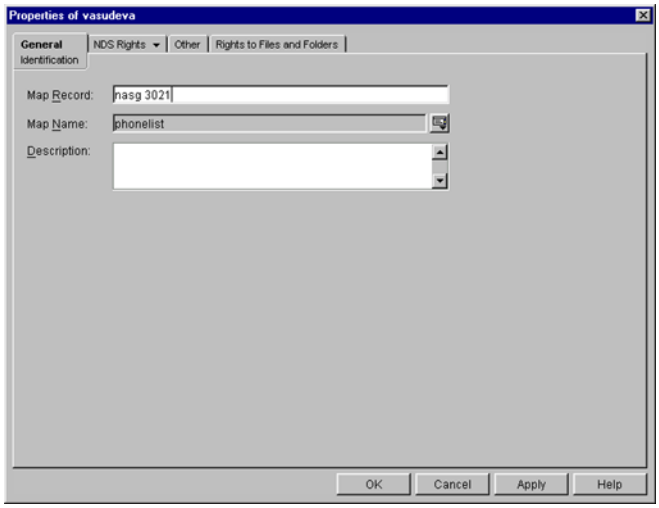
1. In the ONC RPC Number field, enter the RPC number of the program.
2. Enter a brief description of the record.
3. Click Apply > OK.

Figure 8 Services Map Records Properties Panel



1. In the IP Service Port field, enter the port number that this service is available on.
2. In the IP Service Protocol field, enter the protocol used to access the specified service.
3. Enter a brief description of the record.
4. Click Apply > OK.

Figure 9 General Map Records Properties



1. In the Map Record field, specify the map record using the following format:

   key record

2. Enter the map name that the record belongs to.
3. Enter a brief description of the record.
4. Click Apply > OK.

Deleting an NIS Map Using NFSAdmin

1. In the ConsoleOne main menu, right-click the map name in the domain's map list.
2. Click Remove > Yes > OK.

Setting Up a NetWare Server As a NIS Master

1. Create a text file called YPSERV in SYS:\ETC\NIS. For every slave server and enter the hostname of the slave server in this file in the following format:

slaveserverhostname1 slaveserverhostname1

2. Enter the YPSERVERS map entry in SYS:\ETC\NIS\NISMAKE with its path in the following format:

   YPSERVERS SYS:\ETC\NIS\YPSERV

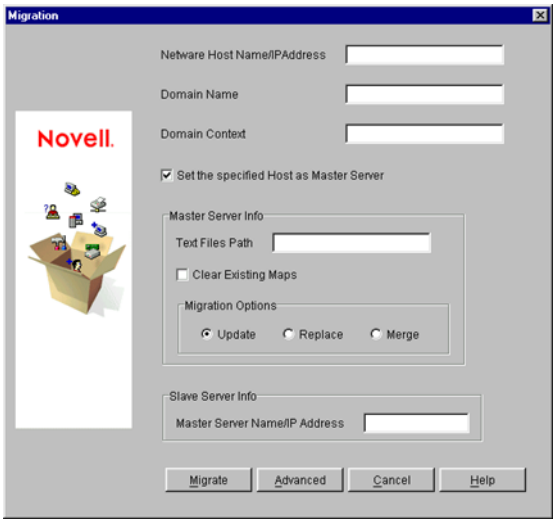
3. If the domain is already migrated, migrate it again to migrate the YPSERVERS map. For migration, see "ConsoleOne Based Management" on page 28.

   If the domain is not already migrated, copy the NIS related text files such as passwd and group (which are available in /etc in UNIX.) into SYS:\ETC\NIS of NetWare server, and then migrate a domain.

4. Load NISSERV.NLM.

Setting up NetWare Server As NIS Slave Server

1. While setting up the UNIX machine as the master, add the NetWare machine name to the slave server list.
2. In the left panel of the ConsoleOne, click The Network.
3. Select the servers' tree where you want to manage the domains and maps.
4. Click the toolbar M icon. The following panel appears.



5. To migrate a domain, enter the NetWare Host Name/IP Address, slave Domain Name, and context where domain object is to be created.
6. To set the NIS Server as slave for this specified domain, uncheck Set the Specified Host As Master Server.
7. Enter the Master Server's Name / IP Address in the Slave server information.
8. To migrate the domain, click Migrate.

NOTE: For the maps Group and Passwd, we do not recommend using the NIS server as a slave. The reason is that in order to serve the NIS client calls of specific domain calls, the slave server looks for the records in all its domains and displays the records of all the domains.

3 Setting Up on Novell Cluster Services

To get the full benefit of using Novell ® Native File Access for UNIX with Novell Cluster Services™, the software must be installed and configured to work in a cluster environment. This chapter deals with the configuration of Native File Access for UNIX on Cluster.

In this chapter the following topics are discussed:

- Prerequisites
- Configuring Load And Unload Scripts
- Component Specific Configuration
- Starting and Stopping Native File Access for UNIX with Cluster Services

Prerequisites

Before installing Native File Access for UNIX with cluster support, create a shared volume and a Cluster Volume object.

1. Create a shared volume using NWCONFIG > NSS volumes.

   NOTE: Do not use the name *nfsclust* because it is a reserved word.

2. To create a Cluster Volume object from the ConsoleOne™ snap-in, complete the following:

   2a Select the Cluster object.

   2b Click File > New > Cluster > Cluster Volume.

   2c Browse and select the shared volume.

   2d Enter the secondary IP address or the virtual IP address associated with the cluster.

The address will be in the following format:

AAA.BBB.CCC.DDD

2e Check the Define Additional Properties check box and click Create.

2f Set the Start, Failover, and Failback Modes.

2g Verify the order of the servers in the nodes list.

2h To save the changes to the Cluster Volume object, click OK.

IMPORTANT: After the shared volume *servername_shared vol name* is cluster enabled, ConsoleOne renames it to *cluster object name_shared vol name*.

ConsoleOne creates a virtual server associated with the shared volume and is called *cluster object name_shared vol name_SERVER*.

ConsoleOne also creates a Cluster Volume object called *shared vol name_SERVER* in the Cluster object container.

3 If the NFS Services are running, run NFSSTOP. Install Native File Access For UNIX on all the nodes in the cluster

4 Delete all the NISSERV_servername objects on NDS.

5 Run nisinst -s *shared vol name_SERVER* from one of the nodes in the cluster

6 In the shared volume which is cluster enabled for the Native File Access for UNIX, make a directory, named ETC.

7 From the node, copy the following files from SYS:/ETC to etc directory of shared volume.

- nfs.cfg
- nis.cfg
- nisserv.cfg
- nfsexprt
- nfsthost
- nfsstart.ncf
- nfsstop.ncf

Configuring Load And Unload Scripts

To customize your specific NetWare NFS Services configuration, edit the IP addresses and volume-specific commands in the load and unload scripts of the

cluster volume object to which you are going to associate NFS Services. Select and right-click the Cluster Volume object and then click Properties to find the Cluster Resource Load Script and Cluster Resource Unload Script. Following are the formats for these scripts.

Load Script

To the load script, add the following at the end of the existing script

```
nfsclust AAA.BBB.CCC.DDD shared vol name shared vol
  name_SERVER
shared vol name: \ETC\NFSSTART
```

Unload Script

To the unload script, add the following at the beginning of the existing script.

```
shared vol name :\ETC\NFSSTOP
```

Setting The Start, Failover, and Failback Modes

With the Start mode set to Auto, the services automatically start on a server when the cluster is first brought up. If the Start mode is set to Manual, you can manually start NFS Services on a server whenever you want.

With the Failover mode set to Auto, services automatically starts on the next server in the Assigned Nodes list in the event of a hardware or software failure. If the Failover mode is set to Manual, you can intervene after a failure occurs and before NFS Services is moved to another node.

With the Failback mode set to Disable, services does not failback to its most preferred node when the most preferred node rejoins the cluster. If the Failback mode is set to Auto, NFS Services does automatically failback to its most preferred node. Setting the Failback mode to Manual prevents NFS Services from moving back to its preferred node when that node is brought back online until you are ready to allow it to happen.

View or change the Start, Failover, and Failback modes by doing the following:

1. On ConsoleOne panel select and double-click the cluster object container.
2. Right-click the cluster resource object *shared vol name*_SERVER and select Properties.
3. Click the Policies tab on the property page.
4. View or change the Start, Failover, or Failback mode.

Component Specific Configuration

The procedure to configure the components of Native File Access for UNIX is much the same as when you configure the components without cluster services. However, some points must be kept in mind while configuring the following components:

- Network File System Server
- Network Information Service

For the location of the configuration files for Native File Access for UNIX with and without Cluster Services, see "Location of Configuration Files" on page 65.

Network File System Server

While configuring the Network File System Server, note the following:

- Export only the shared volumes from NFS Server. For exporting other shared volumes, make sure you mount those volumes in the load script by adding the following before nfsclust entry:

  ```
  nss /activate=volumename
  mount volumename volid = 253 or lesser than this number
  trustmig volumename watch
  ```

  After this, deactivate and dismount them in the unload script

  ```
  trustmig volumename unwatch
  dismount volumename
  nss /deactivate=volumename
  ```

- When mounting from an NFS client, use the virtual IP address of the cluster volume object.

For more information on configuring the Network File System Server, see "ConsoleOne Based Configuration for NFS Server" on page 36.

Network Information Service

While configuring the NIS clients, note the following:

- Bind the NIS clients to NIS server running on cluster using virtual IP address.

Location of Configuration Files

Most of the configuration files are now located in the shared volume's ETC directory. The following table lists the location with and without the cluster services.

Table 3 Location of Configuration Files

Filename	Without Cluster Services	With Cluster Services
NFS.CFG	SYS:\ETC	<Shared Vol Name>:\ETC
NIS.CFG	SYS:\ETC	<shared vol name>:\ETC
NFSSERV.CFG	SYS:\ETC	<shared vol name>:\ETC
NFSEXPRT	SYS:\ETC	<shared vol name>:\ETC
NFSTHOST	SYS:\ETC	<shared vol name>:\ETC
Log file for NFSERV (default is NFSSERV.LOG)	SYS:\ETC	<shared vol name>:\ETC
NISMAKE	SYS:\ETC\NIS	SYS:\ETC\NIS
NFSSTART.NCF	SYS:\SYSTEM	<shared vol name>:\ETC
NFSSTOP.NCF	SYS:\SYSTEM	<shared vol name>:\ETC

Starting and Stopping Native File Access for UNIX with Cluster Services

1. To start NFS Services, from the Cluster ConsoleOne, click Cluster Object > View > Cluster State > Cluster Vol Object Online.
2. To stop NFS Services, from the Cluster ConsoleOne, click Cluster Object > View > Cluster State > Cluster Vol Object Offline.

For additional information on setting up and configuring Novell Cluster Services, see the Novell Cluster Services documentation.

4 Additional Utilities

This chapter discusses some other utilities that are packaged with Novell® Native File Access for UNIX. However, these utilities are not supported.

- SCHINST
- NISINST

SCHINST

This utility is used to extend the schema necessary for storing the UNIX information of objects. The usage is as follows:

schinst [-f *filename*]

The -f *filename* is an optional parameter. It is the name of the file that contains the list of schema files that need to be extended. If filename is not specified, the default file, SYS:\ETC\UNIXSCH is used for extension.

Schinst takes as input the administrator's FDN and password for extending the schema.

Schinst also creates the NISUserDef object with admin equivalent rights and adds the UNIX Profile to root user as UID = 0, GID =1, Home Directory =/home to this object.All log messages generated by schinst are written to the file SYS:\ETC\SCHINST.LOG. All information regarding schema extension will be found in SYS:\SYSTEM\DSMISC.LOG

This utility is run automatically during install. If the directory services are reinstalled or if the NISUserDef object is deleted, run this utility manually.

NISINST

This utility creates an NDS object with name NISSERV_SERVERNAME by default or whatever name was specified with -s option. NIS Server uses this object to store the domains served by the NIS Server. NIS Server validates every request against the list of domains specified in this object. It serves the request only when the domain in the request is present in the above list. The usage of nisinst is:

nisinst [-s *name*] [-x *context*]

The parameter -s is optional, and specifies the name to be given to the object. The parameter -x is also optional and specifes the context where to create this object.

IMPORTANT: If directory services are removed, you need to comment the SEARCH ROOT parameter in NFS.CFG, and do the following:

nfsstop
schinst
nisinst
nfsstart

5 System Messages

This chapter describes the system messages of the various components of Novell Native File Access for UNIX.

MakeNIS

Setting the log file

Possible Cause: The directory specified when creating the log file might be incorrect.

Action: Check for the validity of the directory path you specified.

Required parameters are missing

Explanation: The domain name is mandatory.

Possible Cause: The user has not specified the required parameters.

Action: Enter all the mandatory parameters.

Domain name is missing

Possible Cause: The user has not specified the domain name.

Action: Enter the domain name.

No make data for map

Possible Cause: There is no data corresponding to the map in the makefile.

Action: Enter the record corresponding to the map.

File is older than corresponding map

Possible Cause: The text file used for making this map is older than the map that exists on NDS.

Action: Change the time stamp on the text file by saving it again.

Object with same domain name already exists

Possible Cause: An NDS error occurred while adding the specified object.

Action: Check whether the object already exists.

Unable to add users to group objects

Explanation: Users are already present.

Unable to get the host name or IP address of the machine

Possible Cause: The configuration files containing the host data are not correct.

Action: Check the configuration file.

NIS Installation

Opening configuration file

Possible Cause: Either the file is not present in the specified location or the input is illegal.

Action: Check for the existence of the specified file or the validity of the input.

Reading configuration file

Possible Cause: It is not the correct configuration file.

Action: Check the configuration file.

Getting default host names

Possible Cause: Unable to get the DNS name of the current host.

Action: Check whether entries in relevant configuration files are correct.

Updating the configuration file

Possible Cause: Either the configuration file is not present or it is corrupted.

Action: Check the configuration file.

NIS Services

Internal error with refresh watchdog

Possible Cause: The refresh thread of the NIS Server is failing.

Action: Unload the NISSERV.NLM and load it again.

RPC error

Possible Cause: There was an error on the RPC client call to NIS Server.

Action: Unload the NISSERV.NLM and load it again.

Internal error

Possible Cause: Failure to allocate memory for the domain list of the NIS Server.

Action: Unload the NISSERV.NLM and load it again.

Resource failure

Possible Cause: An NIS Server internal error occurred while allocating memory for its internal structure.

Action: Unload the NISSERV.NLM and load it again.

Unable to allocate space for domain index list

Possible Cause: Failure to allocate memory for the domain list of NIS Server.

Action: Unload the NISSERV.NLM and load it again.

Unable to respond to RPC request

Possible Cause: Failure in sending the RPC response back to the client because of the PKERNAL.NLM.

Action: Repeat the client call.

NFS Server

Unable to determine current server language

Possible Cause: The language of the server is not set.

Action: Set the language for the NetWare server.

Root not mapped to any user

Possible Cause: No NetWare user has UNIX UID set to zero.

Action: Make sure the schema is extended and the UNIX UID of the Superuser (admin) is set to zero.

Failed to register with PKERNEL

Possible Cause: NFS Server registration with Portmapper failed.

Action: Unload PKERNEL and load it again.

SVC_REGISTER failed for program, version

Source: Product Kernel.

Possible Cause: The RPC module cannot register the RPC service program.

Action: Ensure that the program and version number are already registered.

RPC/UDP receive queues are full, packet dropped

Possible Cause: Too many UDP packets have been received and have exceeded the UDP receive queue's capacity in the RPC module. Some UDP packets might be lost. This could be caused by an increase of activities such as a RPC broadcast storm.

A NFS Server Access Modes

This appendix provides information about the NFS Server Access Modes.

Whenever a file or directory is created on the NFS side, the Owner, Group, and Others become the trustees on the NetWare side for NetWare-NFS and NFS-NetWare modes. The trustee rights are determined by the rwx permissions that are set on the NFS side at the time of creation. Executing a chmod command would actually result in the trustee rights getting changed for NFS-NetWare Mode. Apart from changing the trustee rights, the file's attributes are also decided by the rwx permissions. If all the three of them (Owner, Group, Others) have only r–r–r–, then the file would be marked as Read-only.

- **NetWare Mode—**Controls access to the exported NFS directory using NetWare access control methods such as NetWare rights and attributes. This mode is to be used only when the control needs to be on the NetWare side and the volume is exported only for NFS sharing.

  When using this mode, NFS permissions do not modify the settings of the NetWare rights and attributes.

  This mode functions as follows:

  - Trustees are not assigned and attributes are not mapped. [RWX] is not mapped to [SRWCEMFA].
  - By default, IRM is set to SRWCEMFA.
  - The default permissions for files created by UNIX users are based on the effective rights of the mapped NetWare user account.
  - NetWare ownership of a file is determined by the mappings between NetWare and UNIX global objects. The OwnerID (NetWare files) is mapped to the UNIX owner of the file.
  - Files created from DOS by unmapped users have a UID=0.

- The chmod, chown, and chgrp commands have no effect upon the NetWare rights and attributes of the file and they fail silently.
- Executing a chmod command reflects changes on the UNIX side only when the _x (execute permission) is involved.
- The functions of this mode are listed in Table 4

Table 4 NetWare Mode Functions

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
Creation	• Owner ID is mapped to NetWare user. • IRM is set to the default (SRWCEMFA). • No trustees are created. • No attributes are set based on FMode.	• UID is set to Root. (UNIX user is mapped to Admin.) • GID is set to whatever the file was created with. • File mode is set.
• Modification • chown • chgrp • chmod	• No change in the owner ID or trustee. • No change in the group trustee. • No attribute change. • No trustee change. • No IRM change.	• A UID change is not allowed. • A GID change is not allowed. • An Fmode change is effective only for _x.

If, for example, a file is created by user *sara* belonging to group *test*, with **`ls -l`**, the attributes would be as follows:

```
file  rwxr-r-  sara  test
```

The GID is the primary GID of sara, but on the NetWare client side sara is the owner of the file.

The chmod request is successful only for –x–x–x. For example, if *vol1* with a file *fil1* and with only [R———F] for a group is exported, the NFS rights would be r-x.

```
file  r-xr-r-  sara  test
```

Executing the chmod 777 fil command does not change the rights. It still shows as r-x.

file r-xr-xr-x sara test

- **NetWare-NFS Mode—**Creates trustee rights to emulate NFS permissions whenever permissions are created or changed by the NFS client.

  In using this mode, NFS permissions override NetWare rights, but NetWare attributes are always enforced and therefore cannot be overridden by NFS permissions. Use this mode if you want NetWare trustee rights to emulate NFS permissions.

  This mode functions as follows:

  - Up to three NetWare trustee assignments corresponding to the mapped User, Group, and Others are automatically assigned to all files created by UNIX users. The trustee rights correspond to the UNIX user's umask.
  - NetWare ownership of a file is determined by the mappings between NetWare and UNIX global objects. The Owner ID in the DOS name space is the mapped UNIX user on the NFS side.
  - Executing the chmod, chown, or chgrp command from UNIX modifies the trustee assignments to the file, but has no effect on any NetWare attributes. The chmod command works if the owner of the file executes the command from the UNIX side.
  - IRM is set to Others' rights (rights of the organization trustee).
  - Changing the UID/GID/FMode changes the trustees and their rights.
  - The functions of this mode are listed in Table 5.

Table 5 NetWare-NFS Mode Functions

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
Creation	◆ Owner ID - NetWare user is mapped to the NFS user creating the file. ◆ Trustees are created for User/ Group/World mappings. ◆ Trustee rights are set according to rwx. ◆ No attributes are set based on Fmode.	◆ UID/GID is set to whatever the file was created with. ◆ Fmode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r–. chmod + w file1 will fail and permission will remain unchanged as r–r–r–. chmod + x file1 will change permission to r-xr-xr-x.
◆ Modification ◆ chown ◆ chgrp ◆ chmod	◆ Owner ID - User trustee changed to new UID. ◆ Group trustee is changed. ◆ Trustee rights change according to the new rwx. ◆ Attributes do not change.	◆ UID is set to the new UID. ◆ GID is set to the new GID. ◆ Fmode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r–. chmod + w file1 will fail and permission will remain unchanged as r–r–r–. chmod + x file1 will change permission as r-xr-xr-x.
◆ Attributes ◆ DI ◆ RI ◆ RO	◆ DOS user cannot delete file/ directory. ◆ DOS user cannot rename the file. ◆ DOS user cannot write to the file.	◆ NFS user cannot delete file/directory. ◆ NFS user cannot rename the file. ◆ NFS user cannot write to the file.

The User, Group, and Others get mapped to the appropriate NetWare user groups and their rights [rwx] get mapped to [SRWCEMFA], NetWare rights. But the attributes are not affected. For example, if the file were created with r–r–r–, NetWare would not change the attribute to [RO] for that file.

For any file that gets created from the NFS side, or for files that already exist in NetWare when the volume is exported, any change in the permissions would change the trustee rights on the NetWare side.

For example, if a user sara, whose primary group is test, creates a file on the NFS side, the trustees on the NetWare side would be as follows:

.sara.novell	User
.test.novell	Group
.o-novell	Others

The chmod command would work only for a Superuser on the NFS side executing the command on a volume exported with ROOT access. For other users and other non-Root access exports, the command would not succeed.

The chgrp command changes the trustees on the NetWare side. If for a particular UNIX group there is no mapped NetWare group, the NetWare server would be made the trustee.

The chmod command has the effect of modifying the trustee rights appropriately.

- **NFS-NetWare Mode**—Creates trustee rights and NetWare attributes to emulate NFS permissions whenever permissions are created or changed by the NFS client.

  When using this mode, NFS permissions override both NetWare rights and attributes. Use this mode if you want both NetWare trustee rights and attributes to emulate NFS permissions.

  This mode functions as follows:

  - Up to three NetWare trustee assignments corresponding to the mapped User, Group, and Others are automatically assigned to all files created by UNIX users. The trustee rights correspond to the UNIX user's umask.
  - NetWare ownership of a file is determined by the mappings between NetWare and UNIX global objects.
  - Applying the Delete-Inhibit or Rename-Inhibit flag to a file from NetWare does not affect the directory permissions for UNIX users.

- Executing chmod, chown, or chgrp from UNIX can modify both the NetWare trustee assignments and attributes of the file. For example, suppose the following command is executed from a NFS client:

  `chmod 444 file`

  Not only would the write permission be removed from the NetWare trustee assignments of the file, but the Read-only attribute would also be set.

  For example, if the file has been given [RW] permissions on the NetWare side, and on the NFS side the owner of the file executes a chmod command and changes it to r–r–r–, the attributes are changed to [R] on the NetWare side.

  The functions of this mode are listed in Table 6.

Table 6 NFS-NetWare Mode Functions

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
Creation	◆ Owner ID - NetWare user is mapped to the NFS user creating the file. ◆ Trustees are created for User/ Group/World mappings. ◆ Trustee rights are set according to rwx. ◆ Attribute set is based on Fmode.	◆ UID/GID is set to whatever the file was created with. ◆ Fmode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r–. chmod + w file1 will change the permission to rw-rw-rw and remove the RO attribute. chmod + x file1 will change the permission to r-xr-xr-x.
◆ Modification ◆ chown ◆ chgrp ◆ chmod	◆ Owner ID - User trustee is changed to a new UID. ◆ Group trustee is changed. ◆ Trustee rights change according to the new rwx. ◆ The RO attribute is set if the file does not have permission for User, Group, and Other.	◆ UID is set to the new UID. ◆ GID is set to the new GID. ◆ Fmode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r–. chmod + w file1 will change the permission to rw-rw-rw and remove the RO attribute. chmod + x file1 will change the permission to r-xr-xr-x.

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
◆ Attributes ◆ DI ◆ RI ◆ RO	◆ DOS user cannot delete file/ directory. ◆ DOS user cannot rename a file. ◆ DOS user cannot write to the file.	◆ NFS user can delete file/ directory. ◆ NFS user can rename the file. ◆ NFS user can write to the files.

- **NFS Mode**—Does not automatically map between file systems.

  Use NFS Mode if the directory is accessed primarily by NFS clients.

  This mode functions as follows:

  - Trustees are not assigned and attributes are not mapped.
  - UNIX permissions are not mapped to NetWare rights.
  - The default permissions for files created by UNIX users are based on the umask of the user. By default, IRF is set to Null for directories.
  - All files created from UNIX are owned by the NetWare file server when viewed from NetWare.
  - Files created from DOS have a UID=0 and a permission mode of 000. OwnerID of the DOS Name Space is NOBODY.
  - The chmod, chown, and chgrp commands modify the UNIX attributes of a file, but they have no effect on the NetWare rights and attributes of the file.
  - The functions of this mode are listed in Table 7.

Table 7 NFS Mode Functions

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
Creation	◆ OwnerID is set to Nobody. ◆ IRM is set to Null for a subdirectory. ◆ No trustee is created for NetWare User/Group/Other. ◆ No attributes are set according to Fmode.	◆ UID/GID is set to whatever the file was created with. ◆ File mode is set.
◆ Modification ◆ chown ◆ chgrp ◆ chmod	◆ OwnerID is set to Nobody. ◆ No change in group trustee. ◆ No change in attributes, IRM, or trustee rights.	◆ UID is changed. ◆ GID is changed. ◆ File mode is set appropriately.
◆ Attributes ◆ DI ◆ RI ◆ RO	◆ DOS user cannot delete file/ directory. ◆ DOS user cannot rename the file. ◆ DOS user cannot write to the file.	◆ NFS user can delete file/ directory. ◆ NFS user can rename the file. ◆ NFS user can write to the file.

For files that already exist and that were created by NetWare clients, the *owner* and the *group* become Admin and Admin's primary group.

For files that are created from the NFS side, the user group would not get mapped to any trustee on the NetWare side. The owner ID of the DOS name space is set to Nobody, which means that the file is owned by the NetWare server. Some versions of FILER would indicate no owner and older versions would indicate that the NetWare Server owns it.

If the volume is exported with root access for the UNIX machine and the Superuser executes a chown command for an existing NetWare file, the owner ID of the file would change to Nobody. Therefore, an empty directory is exported for NFS Mode.

The file would still be accessible to NetWare users (trustees, etc.). For example, if user sara creates a file from the NFS side, on the NFS side you will see the following:

```
file  rw-rw-rw-  sara  test
```

When the command rights File/t is executed on the NetWare client, the message `no trustees are assigned` is displayed. For any change that occurs, like chmod, NFS name space gets modified.

- **Independent Mode**—An extension of the NFS Mode. Like the NFS Mode, this mode stops access mapping between NetWare and NFS Independent Mode.

  In this mode, access control at NetWare and at UNIX are independent of each other.

  This mode functions as follows:

  - Trustees are not assigned.
  - No mapping of rwx SRWCEMFA is done.
  - No attribute mapping is done.
  - IRM is set to the default (SRWCEMFA).
  - OwnerID of DOS name space is mapped to the UNIX user on the NFS side.
  - Changing Group and Fmode does not affect the DOS side.
  - A file or directory created from the DOS side sets mapped UNIX UID and GID.
  - A file or directory created from DOS has the following default permissions:

    -rw-r-r– (file)

    drwxr-xr-x (directory)

    The functions of this mode are listed in Table 8.

Table 8 Independent Mode Functions

Operations from the NFS Side	DOS Name Space (NetWare Clients)	NFS Name Space (NFS Clients)
Creation	◆ OwnerID - NetWare user is mapped to the NFS user creating the file. ◆ IRM is the default SRWCEMFA. ◆ No trustees are created for User/Group/Other. ◆ No attributes are set according to Fmode.	◆ UID/GID is set to whatever the file was created with. ◆ File mode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r– for the first time only. chmod + w file1 will change the permission to rw-rw-rw-. chmod + x file1 will change the permission to r-xr-xr-x.
◆ Modification ◆ chown ◆ chgrp ◆ chmod	◆ OwnerID changes. ◆ No change in group trustee. ◆ No change in attributes, IRM, or trustee rights.	◆ UID is changed. ◆ CID is changed. ◆ File mode is set as follows: If file1 has the RO attribute set, its permission will be r–r–r– for the first time only. chmod + w file1 will change the permission to rw-rw-rw-. chmod + x file1 will change the permission to r-xr-xr-x.
◆ Attributes ◆ DI ◆ RI ◆ RO	◆ DOS User cannot delete file/directory. ◆ DOS User cannot rename the file. ◆ DOS User cannot write to the file.	NFS User can delete file/directory. NFS User can rename the file. NFS User can write to the file.

Comparing NetWare and NFS File Security

NetWare and NFS file and directory security differ in some respects, but both systems base the degree of security on the following:

- A user's need for and right to the information

- The sensitivity of the information

To protect file information, the system separates users into classes. Each class is permitted only necessary file access.

When an exact mapping of rights between the two systems is not possible and the access control mode specifies a conversion, the file sharing service translates the rights in favor of tighter rather than looser control. Even though the automatic translation between NetWare rights and NFS permissions honors the security of both systems, the following can occur:

- A user can have rights in NetWare that are denied in NFS because a direct translation is not possible. It is better to limit permissions than grant unintended access.
- Rights added from NetWare might not add to the permissions allowed for the NFS user.

The automatic mapping of access controls has the following advantages:

- The system administrator can control the trade-off between security and ease-of-use.
- It maintains each client system's level of security.
- It does not impair the operation of other software.
- It functions simply, and the process implements nothing new beyond the normal requirements of NetWare or UNIX. A NetWare user or a UNIX user does not need to know anything about the access rights in the other system.

The following sections discuss the NFS and NetWare file access controls.

NFS Controls

For each of the three NFS user classes (User, Group, and World), there are three access controls, called *permissions*. For a file, these permissions allow a user to read from, write to, and execute the file.

For a directory, the same permissions apply. Users need read permission to use the ls command to list the files in a directory. They need write permission to add or remove files from a directory. They need execute permission to access the directory with the cd command or use the directory as part of a path. To access a file in a subdirectory, users must have the proper permissions for the file and for all the directories in the path.

NetWare Controls

NetWare Rights Security is based on the combined effects of trustee rights and rights invoked with the Inherited Rights Mask. (See the NetWare documentation for descriptions of these NetWare security terms.) The actual rights a user can exercise in a directory or file depends on these combined rights, which are referred to as a user's *effective rights*. Effective rights translate between NFS and NetWare directories and files.

The NetWare effective rights that pertain to either directories or files are summarized in the following table.

Table 9 Description of NetWare Rights

NetWare Rights	Privileges Granted
Supervisor	All rights; overrides any restrictions placed by the Inherited Rights Mask.
Read	Right to open and read or execute.
Write	Right to open and modify.
Create	Right to create; when assigned to a file, allows a deleted file to be recovered.
Erase	Right to delete.
Modify	Right to rename a file and to change attributes.
File Scan	Right to see directory or file listings.
Access Control	Right to modify trustee assignments and the Inherited Rights Mask.

Effective rights can also be transferred from one user to another. This transfer of effective rights, called *security equivalence,* allows one user to have the same rights as another. Transferred effective rights are also translated between NFS and NetWare.

Besides a user's effective rights, some NetWare file attributes put additional controls on specified directories or files. (Refer to Attributes in the NetWare documentation.) These controls take precedence over a user's effective rights. A user with the Modify right, however, can override the file attributes.

Impact of NetWare Security on NFS

If the NFS Gateway file sharing service is set up to use an access mode that translates access controls, the file sharing service effectively honors NetWare security on all given files and directories. In doing so, the rights as seen from NFS might appear more restrictive than the rights as seen from NetWare DOS. This apparent discrepancy occurs because NFS permissions are not as comprehensive as NetWare rights, and NFS might have no way of representing a right that is present on NetWare.

Impact of NFS Security on NetWare

If the NFS Gateway file sharing service is set up to use an access mode that translates access controls, NFS access can become too restrictive. If this happens, consider the following options:

- Relax your NFS security by using a different access control mode.
- Store data that users need to share in separate directories and files, and apply fewer restrictions.

Converting NetWare Attributes to NFS

When you convert NetWare directories and files to NFS, NetWare file attributes and UNIX permissions interact and specific NetWare attributes affect UNIX permission settings. You can control this interaction by setting the Access Control mode.

NetWare Attributes Control

NetWare confines a user's access to directories and files with equivalent rights derived from trustee rights and the Inherited Rights Mask. To further protect certain directories and files, NetWare uses attributes such as Delete Inhibit. These attributes determine what you can do to a file or directory. These attributes take precedence over NetWare user rights, even the inherent rights of an administrator.

NFS permissions map satisfactorily to trustee rights. Attributes, however, do not correspond directly to NFS permissions. It is not certain whether the setting of an NFS permission will change a NetWare attribute. Also, a security-safe conversion of some attributes can be too restrictive on the NFS file. For example, some attributes imply that users cannot remove the file.

Implementing this on the NFS side requires revoking the write permission at the directory level.

Because of these mapping ambiguities, Novell Native File Access for UNIX gives you a choice, allowing you to trade security for flexibility. For a given file system, you can determine whether the assignment of NFS permissions should affect a NetWare file attribute and whether in some situations NFS can override protections implied by attributes. You make the choice by setting the access control mode.

When you select NetWare Mode and NetWare-NFS Mode, setting permissions from NFS does not affect the settings of the NetWare attributes. However, the NetWare Read-only attribute affects the permissions presented to the NFS user by revoking write permission from a file. This mode enforces the complete protection implied by NetWare attributes.

When you select the NFS-NetWare Mode, the attributes of a file do not result in changes to directory ownership or directory permission. In some cases, changes to permissions from NFS result in changes to the attributes of a file. NFS-NetWare Mode is the easiest method for sharing files between NetWare clients with the least effect on security.

The following sections describe how specific NetWare attributes are converted, depending upon the setting of the access mode, first for files and then for directories.

Converting File Attributes

The NetWare NFS Server maps file attributes according to the access mode as follows.

When you select NetWare Mode or NetWare-NFS Mode, the NetWare NFS Server converts NetWare file attributes.

Table 10 NetWare and NetWare-NFS Mode File Attribute Conversion

NetWare Attribute	NFS Conversion	Notes
Delete Inhibit	The permissions and owner of the file and the parent directory remain unaffected. If UNIX user tries to delete this file, the operation will fail, indicating that permission is denied.	If the file system allows root access, the Superuser on the client system can delete any file or directory in the parent directory except the ones with the Delete Inhibit attribute.
Read-only	The file's permission changes to r–r–r– . Owner will remain unaffected. Write operation will fail, indicating that permission is denied.	
Rename Inhibit	Functions the same as Delete Inhibit.	
Transactional	The chmod operation will fail.	

When you select NFS-NetWare Mode (default mode), the NetWare NFS Server converts the NetWare file attributes.

Table 11 NFS-NetWare Mode File Attribute Conversion

NetWare Attribute	NFS Conversion
Delete Inhibit	Makes no change to permissions. UNIX user can delete the file.
Read-only	Makes the file unwritable. Resetting the write permission on the file with the chmod command removes the Read-only attribute. When the NFS permission is set to read only for Owner, Group, and Others, the Read-only attribute is set on the NetWare file. This setting results in the file being seen as read-only from DOS and locked from a Macintosh* client.
Rename Inhibit	Functions the same as Delete Inhibit.

When you select the NFS Mode, NFS permissions do not affect NetWare attributes and NetWare attributes do not affect NFS permissions.

Converting Directory Attributes

When you select NetWare Mode or NetWare-NFS Mode, the NetWare NFS Server converts the NetWare directory attributes.

Table 12 NetWare and NetWare-NFS Mode Directory Attribute Conversion

NetWare Attribute	NFS Conversion
Delete Inhibit	Changes the UID of the parent directory's owner to 0 when viewed from NFS. Revokes write permission for the parent directory for User, Group, and Other. This is similar to Delete Inhibit for files.

When you select NFS-NetWare Mode, the NetWare NFS Server converts the NetWare directory attributes.

Table 13 NFS-NetWare Mode Directory Attribute Conversion

NetWare Attribute	NFS Conversion
Delete Inhibit	Makes no change to permissions. Resetting the write permission of the parent directory with the chmod command removes the Delete Inhibit attribute from all files within the parent directory.

When you select NFS Mode, NFS permissions do not affect NetWare attributes and Other Characteristic Translation.

Other NetWare file characteristics translate to NFS as follows.

NetWare Attribute	NFS Conversion
Owner ID	The NetWare login name changes to the NFS UID according to the setting in the user list. An exception occurs whenever the dos_attributes parameter is set to nomodify and the NetWare file attributes Read-only and Delete Inhibit are associated with the file or directory. Specifically, the owner of a file or directory translates to 0 when the following associations exist: ◆ The file has the Read-only attribute. ◆ A file below the owner's directory has the Delete Inhibit attribute. ◆ The file's owner does not have the Access Control right.
File Size	The file size is maintained in bytes.
Create Date and Time	The date and time are maintained as in DOS. NFS specifies time in number of seconds since January 1, 1970. NetWare converts Create Date and Time to NFS format for the NFS ctime attribute.
Last Update Date and Time	Last update date and time are converted to NFS format for the NFS mtime attribute (similar to Create Date and Time).
Last Access Date	Last Access Date is converted to NFS format and uses 12:00 a.m. as the approximate last access time. NetWare keeps track of the date of last access only. The NFS atime attribute requires date as well as time.

NetWare Rights and UNIX Permissions

This section contains the following topics:

Mapping NetWare Rights and UNIX Permissions

When a user accesses a file on a mounted file system, the request can pass through a NetWare security check, an NFS security check, or both depending on the access mode selected. Each of these security checks function independently. If the access mode specifies security checks on both sides, the Gateway first checks the user's NetWare access rights. Then, if the Gateway accepts the user's request, the request passes to the remote NFS server, and NFS does its check. This arrangement lets the administrator on the NetWare side impose greater restrictions on access control than those set on NFS.

When the Gateway translates NetWare rights to NFS permissions or permissions to rights, as dictated by the access mode, the conversion is nearly equivalent, but a direct one-to-one match is not possible. NetWare file security is more complex and powerful than NFS file security. The method of translating permissions to rights will, if necessary, adjust toward greater restriction rather than lesser in order to preserve the degree of NFS restrictions.

For example, suppose a NetWare administrator grants a Gateway user more rights than the user is permitted on the NFS file. In this case, the permissions on the NFS side do not change to allow more access. Even if the Gateway passes the user's request on to the NFS Server, the NFS Server would still deny access to the file.

The following tables show how NetWare rights and UNIX permissions translate.

IMPORTANT: Where more than one right or permission is shown for a given condition in the following tables, those rights or permissions work in combination. For example, it is the combination of the NetWare rights of Create, Erase, and Write on a directory that translate to the write permission on the NFS side.

Translating NetWare Rights to UNIX Permissions

Table 14 shows how NetWare rights translate to NFS permissions. These conversions happen when you add or delete trustees using NetWare utilities such as FILER.

Translation occurs only when specified by the access mode.

Table 14 How NetWare Rights Translate to NFS Permissions

NetWare Rights	NFS Permissions
Directory	*Directory*
File Scan + Read	Read + Execute
Create + Erase + Write	Write
File	*File*
Read	Read
Write	Write

Table 15shows how NetWare attributes translate to NFS permissions. These conversions happen when you modify a directory entry using NetWare utilities such as FILER and FLAG.

Table 15 How NetWare Attributes Translate to NFS Permissions

NetWare Attributes	NFS Permissions
File	*File*
Read-Only	Removes Write from owner, group, and world
Read/Write	Restores NFS mode that existed prior to NetWare change and adds Read and Write for owner

Table 16 shows how NFS permissions translate to NetWare attributes.

These conversions happen when you create a directory or a file, or when you reference a directory or a file for the first time.

Table 16 How NFS Permissions Translate to NetWare Attributes

NFS Permissions	NetWare Attributes
Directory	*Directory*
no write	Rename Inhibit + Delete Inhibit

NFS Permissions	NetWare Attributes
write	Removes Rename Inhibit + Delete Inhibit
File	*File*
no write	Read Only + Rename Inhibit + Delete Inhibit
write	Removes Read Only + Rename Inhibit + Delete Inhibit

Table 17 shows how NFS permissions translate to NetWare rights.

Table 17 How NFS Permissions Translate to NetWare Rights

NFS Permissions	NetWare RIghts
Directory	*Directory*
read + execute	Read + File Scan
write	Create + Erase + Write + File Scan
File	*File*
read	Read + File Scan
write	Write + File Scan

Permission Mapping

When the NFS Server file sharing service maps NFS permissions to NetWare rights, the original NFS permissions are still retained on the NFS system. This approach is necessary for NFS file and directory access and to simplify reverse mapping. Original information about a user's permissions is retained on the NFS system in cases where the NFS permissions do not have equivalent NetWare rights, such as the execute permission on files.

The NFS Server translates NFS access permissions as follows:

1. First, the NFS user and group information converts to the corresponding NetWare user and group information.
   - The NFS user identification number converts to the corresponding NetWare username as indicated in the user list. If the user

identification number is not in the list, the NetWare NDS Server object is assigned.

- The NFS group identification number converts to the corresponding NetWare group name as indicated in the group list. If the group is not in the list, the NetWare NDS Server object is assigned.
- The NFS other or world identification number converts directly to the default OU. (Initially, the world group contains all users in the default OU; however, you can use ConsoleOne to add users to world by importing them from other locations in the NDS tree.)

2. Second, the NFS permissions for each directory's or file's user type are translated to the mapped rights for the corresponding NetWare user type as described in the following section, NFS Permissions to NetWare Rights Translation.

NFS Permissions to NetWare Rights Translation

You can enforce UNIX-style NFS permissions by creating corresponding NetWare trustee rights. In UNIX, every file and directory is assigned an explicit set of permission bits. In NetWare, explicitly setting NetWare trustee rights for each file is not necessary and generally is not done. Trustee rights propagate down the directory structure until they are reset by another trustee right. Consequently, you must choose between administering rights in a way that seems natural from NetWare and emulating UNIX access control.

Whether trustee rights are set through NFS is determined by the access mode. The default mode is NFS-NetWare Mode.

When the chosen access mode does not specify that trustee rights reflect the actions of NFS, permissions cannot be changed by the actions of NFS clients. For example, if the selected mode is the NetWare Mode and a UNIX user attempts to use the chmod command to change the permissions on a file, the command fails silently and no error is returned. However, applications continue to run, because the file sharing service does not return errors for the now ineffective operations on permissions.

Table 18 Translating NFS File Permissions to NetWare

NFS Permission	NetWare Right
read	Read
write	Write

NFS Permission	NetWare Right
execute	Not applicable The execute permission has no direct equivalent in NetWare access controls. The execute permission by itself would not permit the file to be read or written to and, therefore, is not used.

Table 19 Translating NFS Directory Permissions to NetWare

NFS Permission	NetWare Right
read	File Scan and Read rights are granted only if the directory also has NFS execute permission.
write	Erase, Create, and Write for directory The NFS directory write permission allows renaming of files. The NetWare right that allows renaming of directories and files is the Modify right, but this right also allows changing of file attributes. Rather than permit the NetWare user to change file attributes with the Modify right, the Modify right is not granted and the NetWare user, therefore, is not automatically granted the ability to rename NFS directories and files.
execute	File Scan and Read rights are granted only if the directory also has NFS read permission.

In addition to direct translation of these listed permissions, the owner of a file is also assigned the NetWare Access Control right because this right is inherent to ownership of an NFS file. Conversely, denial of the Access Control right from the NetWare side revokes ownership as viewed from NFS.

When the UNIX Superuser uses the chown command to change the NFS ownership of a file, the equivalent NetWare user is granted Access Control to the file.

Rights Propagation

Trustee rights assigned to a particular user or group for a directory are propagated to all the files within that directory. The only exception is if the file has assigned the same trustee, then that trustee rights overrides the inherited rights.

NetWare Equivalent Rights to NFS Permissions Translation

When you access a NetWare file from NFS or change an NFS file's access control from NetWare, the equivalent rights for the NetWare file owner or for the NetWare group are translated to NFS permissions. All other NetWare equivalent rights are ignored.

When a file's trustee assignments are changed, the NetWare rights are converted to NFS permissions as follows:

- For every trustee assignment, the owner and group effective rights convert to corresponding NFS permissions.
- The effective rights for the NetWare group mapped to world convert to the NFS permissions designated for the user class other.

The NetWare classes are converted to their equivalent NFS classes in accordance with a mapping table. If the trustee assignment changes for a directory, the conversion propagates to the files under that directory that have Inherited Rights Masks set to allow the change.

Table 20 Translating NetWare File Rights to NFS

NetWare Right	NFS Permission	Notes
Read	read	
Write	write	
Create	Not applicable.	Create, generate, and scan file commands do not belong in the NFS file.
Erase	Not applicable.	Create, Erase, and File Scan do not apply to NFS files. These rights are converted to write and execute permissions for the parent directory.
Access Control	No direct match.	The Access Control right is a prerequisite to file ownership.
File Scan	Not applicable.	Create, Erase, and File Scan do not apply to NFS files. These rights are converted to write and execute permissions for the parent directory.
Modify	Not applicable.	

NetWare Right	NFS Permission	Notes
Supervisor	Equivalent to Superuser.	

Table 21 Translating NetWare Directory Rights to NFS

NetWare Right	NFS Permission	Notes
Read	The read permission is propagated to all files under the directory if inheritance permits.	
Write	The write permission is propagated to all files in the directory if inheritance permits.	
Create	The write permission is granted only if the NetWare directory also has the Erase right.	If a directory has both Create and Erase rights, they are mapped to write permission. If the directory has only the Create or the Erase right, this right is dropped when viewed from the NFS.
Erase	The write permission is granted only if the NetWare directory also has the Create right.	If a directory has both Create and Erase rights, they are mapped to write permission. If the directory has only the Create or the Erase right, this right is dropped when viewed from the NFS side.
Access Control	No direct match.	The NetWare owner of the file has the same rights in NFS as the NFS owner of the file. If the NetWare owner of the file does not have the Access Control right, the NetWare owner's identification is mapped to a special NFS identification (UID 0), which does not allow the permissions to be changed from NFS.
File Scan	read, execute	The File Scan right is mapped to read and execute permissions only if all files and subdirectories in the specified directory also have the File Scan right.
Modify	Not applicable.	

NetWare Right	NFS Permission	Notes
Supervisor	Equivalent to Superuser; translation not applicable.	

The following example illustrates how NetWare rights are converted to NFS permissions. This example assumes that user JOHN has security equivalent to user MARY.

User JOHN	R W
User MARY	R W M A S
Group ENGINEERING	R
Group ACCOUNTING	R W
Group EVERYONE	None

Entering the following UNIX command

```
ls -l abc.txt
```

displays the following NFS rights:

```
-rw-r—– 1 john engineering 216 Feb 5 1994 abc.txt
```

NFS owner john (equivalent to NetWare owner JOHN) has read and write permission. NFS group engineering (equivalent to NetWare group ENGINEERING) has read permission. All other NFS users have no permissions, because the equivalent NetWare group (the default OU) has no rights to the file.

NetWare user MARY, who is not the owner but who has NetWare trustee rights, is dropped in the translation. The same is true of the NetWare group ACCOUNTING.

Permissions Guidelines

In general, to avoid confusion, it is best to set up permissions and rights so as not to display files to users on the other systems who cannot use the files. Specifically, when storing files that NFS users access, you can avoid problems by following two rules:

- Do not store applications in directories shared by NFS users, or make sure the application files do not have execute permission.
- Do not store data files created by applications in shared directories unless you know the files are compatible with a version of the application available to NFS users.

Accessing a Service as a User or Member of a Group

When a user attempts to access a remote file system using the NFS Server, the local NFS client sends the user's request to the remote server where the file resides. The server checks whether the user has the necessary permissions and is recognized as one the following:

- The owner of the file
- A member of the file owner's group
- Any user able to log in to the server

To identify those NetWare users and groups on the system without a recognized account, the server maps these accounts to nobody (UID -2) and nogroup (GID -2). This allows the NFS server to identify nonmapped usernames and set special restrictions.