White Paper

Configuring NFSv4 on EMC® VNX® Systems

Abstract

This white paper is an introduction to the NFSv4 protocol on the EMC® VNX™ storage systems. It describes basic configuration of NFSv4 and provides details for using it with LINUX and Solaris. Troubleshooting information is also included.

September 2012

EMC²

Part Number H10949

Table of Contents

Executive summary

This white paper provides an introduction to configuring the EMC® VNX™ storage system to be used with NFSv4. It explains the Data Mover configuration along with host-side best practices for mount options. This white paper describes in detail how to convert the file systems to Unicode.

NFSv4 ACLs and locking are explained along with examples. A troubleshooting section is provided to assist if something is not working after initial configuration.

Audience

This white paper is intended for EMC customers, partners, and employees who are considering the use of NFSv4 on the VNX storage systems. It assumes familiarity with NFS and VNX storage systems and the management software.

Introduction

Network File System version 4 (NFSv4) is a version of NFS with features such as strong authentication and integrity by using Kerberos and SPKM-3, improved performance, lock migration, UTF-8, ACLs, and better support for Windows file-sharing semantics.

Version 4 (RFC 3010, December 2000; revised in RFC 3530, April 2003), influenced by AFS and CIFS, includes performance improvements, mandates strong security, and introduces a stateful protocol. Version 4 became the first version developed with the Internet Engineering Task Force (IETF) after Sun Microsystems handed over the development of the NFS protocols.

With NFSv4, the IETF provides the first openly defined file system protocol. NFSv4 draws upon previous versions of NFS along with characteristics of other distributed file systems to provide a useful, flexible framework for today's client and server environments. NFSv4 provides strong security through the use of Kerberos V5, SPKM-3, or LIPKEY. NFSv4 combines the previously disparate set of protocols surrounding NFS into a single protocol. NFSv4 also allows for adaptation to future needs using minor versioning.

The VNX implementation of NFSv4 supports:

- NFSv4 over the Internet Protocol v4 (IPv4) and v6 (IPv6)
- UNIX and Kerberos v5 for user authentication
- UTF-8 and ASCII
- Pseudo-root file systems
- Access control lists (ACLs)

- Nested mount file systems

Limitations

The VNX implementation of NFSv4 does not support:

- Blocking locks feature
- Access control that uses Windows SIDs
- Public key-based authentication (SPKM-3 and Lipkey)
- ASCII code pages
- server_export -name option – This option adds a name to an NFS export aka alias.
- Log messages that indicate NFSv4 service status (starting and stopping) written to the Data Mover's server log
- Directory delegation

Data Mover configurations

Enable NFSv4 on the VNX system

In VNX OE for File 7.1 and above, NFSv4 is enabled by default, but not started automatically. You no longer have to modify the config file, instead, use the server_nfs command to start it.

```
$ server_nfs <movername> -v4 -service –start
 where: <movername> = name of the Data Mover
```

Note: on 7.1 and above the config file doesn't need to have hivers=4 option.

Prior to the VNX 7.1 release, you need to enable NFSv4 support on the VNX.

1. To enable NFSv4 support on VNX use a text editor, open the /nas/server/slot_‹x›/config file, where ‹x› is the slot number of the Data Mover.

```
$ vi /nas/server/slot_2/config
```

2. Find the nfs config line and append it with the option hivers=4 or, if the option already appears, be sure the value is set to 4. Save and close the file.

```
$ cat /nas/server/slot_2/config
nfs config hivers=4
```

Note: Do not change any other lines or options in the file.

3. Restart the Data Mover by using this command syntax:

```
$ server_cpu <movername> -reboot -monitor now
where: <movername> = name of the Data Mover
```

Configure the NFSv4 domain

You must configure the NFSv4 domain. The NFSv4 domain name is not related to DNS, NIS, or LDAP domains, but it can be the same as the local DNS, NIS, or LDAP domain name or it can be unique for NFSv4. The NFSv4 domain is used only for user and group mapping. For troubleshooting and easier understanding, EMC recommends the NFSv4 domain to be different from DNS, NIS, or LDAP.

To configure the NFS domain:

- Specify the NFSv4 domain name.
- Verify the domain parameter.
- List the NFSv4 parameters.

Specify the NFSv4 domain name

Use the domain parameter to specify the NFSv4 domain name by using this command syntax.

```
$ server_param <movername> -facility nfsv4 -modify domain -value
<new_value>
where:
<movername> = name of the specified Data Mover
<new_value> = name of the domain
```

Example:

To set the domain name to nfsv4.domain, type:

```
$ server_param server_2 -facility nfsv4 -modify domain -value nfsv4.domain
Output:
Server_2 : done
```

Verify the domain parameter

To verify that the domain parameter is set, use this command syntax:

```
$ server_param <movername> -facility nfsv4 -info domain
where:
<movername> = name of the Data Mover
```

Example:

To verify whether the domain parameter was set, type:

```
$ server_param server_2 -facility nfsv4 -info domain
Output:
server_2 :
name             = domain
facility_name    = nfsv4
default_value    = ''
current_value    = ' nfsv4.domain '
configured_value = nfsv4.domain
user_action      = none
change_effective = immediate
```

```
range            = '*'
description      = Sets the NFS v4 domain
```

List the NFSv4 parameters

To list all the NFSv4 parameters with their default, current, and configured values, use this command syntax:

```
$ server_param <movername> -facility nfsv4 -list
where:
<movername> = name of the Data Mover
```

Example:

To list all the NFSv4 parameters, type:

```
$ server_param server_2 -facility nfsv4 -list

Output:
server_2:
param_name         facility   default   current    configured
leaseDuration       nfsv4       40       40
recallTimeout       nfsv4       10       10
domain nfsv4        nfsv4.domain
vnodePercent        nfsv4        80       10         10
32bitClient         nfsv4        1         1
```

Mount a file system for NFSv4 access

Use the server_mount command to mount a file system for NFSv4 access. A mount point must begin with a forward slash (/).

The -option argument on the server_mount command specifies a number of different options for a mounted file system accessed by an NFSv4 client, including:

- Type of access control and ACL management
- Delegation mode

To mount a file system on a Data Mover, use this command syntax:

```
$ server_mount <movername> -option <options> <fs_name> /<mount_point>
where:
<movername> = name of the Data Mover
<options>   = specifies mount options, separated by commas
<fs_name>   = file system to be mounted
<mount_point> = path to the mount point for the Data Mover
```

Example:

To mount a file system on server_2 and set the access policy to MIXED for file system ufs1, type:

```
$ server_mount server_2 -option accesspolicy=MIXED ufs1 /ufs1
```

To mount a file system on server_2, disabling read-write delegation for file system ufs1, type:

```
$ server_mount server_2 -option nfsv4delegation=NONE ufs1 /ufs1

Output:
Server_2 : Done
```

Translate to MIXED access policy

If the filesystem is already populated with data, and mounted with a default NATIVE access policy, it needs to be translated to MIXED access policy.

Note: Access policy conversion can be done online, without disrupting user access.

To translate the filesystem access policy, use these steps:

```
$ nas_fs -translate <fs_name> -access_policy start -to MIXED -from
<previous access policy>
where:
<fs_name>   = file system to be mounted
<previous access policy> = the access policy that the FS was mounted with
before being remounted with MIXED. If access policy was not specified, then
it was NATIVE. Other options could be: NT, UNIX, and SECURE.
```

Example:

To mount a file system on server_2 and set the access policy to MIXED for file system ufs1, type:

```
$ nas_fs -translate ufs1 -access_policy start -to MIXED -from NATIVE
<...output abridged...>
<...output will be the same as with nas_fs -info <fsname>...>
```

To check the status of conversion, use this command:

```
$ nas_fs -translate testfs -access_policy status

status=In progress
percent_inode_scanned=95
```

When conversion is complete, the status will return N/A

```
$ nas_fs -translate testfs -access_policy status

status=N/A
```

Type of access control and ACL management

Table 1 Access Policy

Access-Checking policy	Description
NATIVE (default)	• Access from UNIX or UNIX FTP -- UNIX mode bits • Access from CIFS or CIFS FTP -- Windows ACLs • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change has no effect on Windows ACLs

	• Windows ACL change has no effect on UNIX mode bits
NT (deemed confusing, not recommended)	• Access from UNIX or UNIX FTP -- UNIX mode bits and Windows ACLs • Access from CIFS or CIFS FTP -- Windows ACLs • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change has no effect on Windows ACLs • Windows ACL change has no effect on UNIX mode bits
UNIX (deemed confusing, not recommended)	• Access from UNIX or UNIX FTP -- UNIX mode bits • Access from CIFS or CIFS FTP -- Windows ACLs and Unix mode bits • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change has no effect on Windows ACLs • Windows ACL change has no effect on UNIX mode bits
SECURE (most secure, somewhat difficult to maintain)	• Access from UNIX or UNIX FTP -- UNIX mode bits and Windows ACLs • Access from CIFS or CIFS FTP -- Windows ACLs and Unix mode bits • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change has no effect on Windows ACLs • Windows ACL change has no effect on UNIX mode bits
MIXED (required for NFSv4, easy to change from native)	• Access from UNIX or UNIX FTP -- Windows ACLs • Access from CIFS or CIFS FTP -- Windows ACLs • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change causes Windows ACLs to be rebuilt and overwritten based on new UNIX mode bits • Windows ACL change causes UNIX mode bits to be rebuilt and overwritten based on new Windows ACLs • Windows ACLs are more granular than UNIX mode bits, consequently not all permissions set in an ACL can be translated to UNIX mode bits. In some cases, the UNIX mode bits might show more permissions than a user actually has.

MIXED_COMPAT (deemed confusing, not recommended)	• Access from UNIX or UNIX FTP -- depends on which protocol was last used to set file security • Access from CIFS or CIFS FTP -- depends on which protocol was last used to set file security • Maintained file permissions -- Windows ACLs and UNIX mode bits • UNIX mode bit change causes Windows ACLs to be rebuilt and overwritten based on new UNIX mode bits, however Windows ACLs are not used • Windows ACL change causes UNIX mode bits to be rebuilt and overwritten based on new Windows ACLs, however UNIX mode bits are not used • Windows ACLs are more granular than UNIX mode bits, consequently not all permissions set in an ACL can be translated to UNIX mode bits. In some cases, the UNIX mode bits might show more permissions than a user actually has.

Note: The MIXED policy translates the UNIX ownership mode bits into three Access Control Entries (ACEs): Owner, Group, and Everyone, which can result in different permissions for the Group ACE and the Everyone ACE. The MIXED_COMPAT policy does not translate a UNIX Group into a Group ACE. The Everyone ACE is generated from the UNIX Group.

Note: When accessed from a Windows client, ACLs are only checked if the CIFS user authentication method is set to the recommended default, NT. This is set using the -add security option in the server_cifs command.

Delegation mode

When using NFSv4, the VNX can delegate specific actions on a file to a client, such as more aggressive client caching of data, metadata, and locking. Delegation improves network performance by allowing NFS clients to buffer file data and metadata locally and perform operations on that data and metadata before sending it to the server.

Delegation is configured per file system and read-write delegation is on by default.

EMC recommends turning delegation off when:

- The data needs to be shared frequently by applications on different clients.
- The data is accessed by mission-critical, transaction-based applications, such as databases, where a client failure could impact data integrity.

Note: Because all data operations are executed by the NFSv4 client and not sent to the server during the life of the delegation, the UNIX application on the server is not aware of changes stored by a client. If a client fails, all changes to the data might be lost.

The VNX supports the following file delegation levels:

- None—No file delegation is granted
- Read—Only read delegation is granted
- Read/Write—Read and write delegation is granted

Specify NFSv4 access

Normally a file system is exported to all versions of the NFS protocol. Access to a file system can be restricted by setting the nfsv4only option.

When a file system is exported with the nfsv4only option, but the NFSv4 service is not enabled by the server_nfs command, the problem is logged and the export continues. Although the export does not fail, access through NFSv2 and NFSv3 as well as NFSv4 is blocked.

To export VNX file systems for NFSv4only access, use this command syntax:

```
$ server_export <movername> -Protocol nfs [-option <options>] <pathname>
where:
<movername> = name of the Data Mover
<options>   = options to be applied to the NFS export
<pathname>  = NFS export pathname
```

Example:

To export a file system for NFS access, type:

```
$ server_export <movername> -Protocol nfs -option nfsv4only /ufs1

where:
<movername> = name of the Data Mover
```

Display NFSv4 clients

To display all client systems that have an established state with the NFSv4 server, type:

```
$ server_nfs <movername> -v4 -client -list

where:
<movername> = name of the Data Mover
```

Example:

To display all client systems that have an established state with the NFSv4 service on server_2, type:

```
$ server_nfs server_2 -v4 -client -list
```

```
Output:
server_2 :
------ nfsv4 server client list ------
Hostname/ip:  Index
win901234   :  0xe2400000
10.171.2.76:  0xef400000
```

The first column in the output displays the NFSv4 client hostname or IP address. The second column displays an index number that VNX uses to identify the client connection.

The client list is based on the number of times an NFSv4 client sets a client ID. It notifies the server of its intention to use a particular client identifier, callback, and callback identifier for subsequent requests that entail creating lock, share reservation, and delegation state on the server. Some NFSv4 clients use one client ID for all users and files, but other NFSv4 clients use one client ID per process. Only files can carry state, not directories. NFSv4 clients do not have to maintain an active state. A client's state remains active only as long as the client renews its lease. If a client no longer is in an established state, it is no longer listed in the command output but this does not indicate a problem.

Unicode

For detailed information about Unicode and conversions please refer to Using *International Character Sets on VNX™ for File* document.

NFSv4 requires UTF-8 (also known as Unicode) for character encoding. By default, a system is configured to use ASCII Latin-1 character encoding. Before accessing data through NFSv4 clients, you must change the character encoding method to UTF-8.

Typically, Unicode is enabled during VNX installation. If however, your VNX does not have Unicode enabled, it is easily fixed for any active datamover.

To check whether Unicode is enabled, run this command:

```
$ /nas/sbin/uc_config -info
```

Example:

```
$ /nas/sbin/uc_config -info
Output:
Common filesystem [root_fs_common] exists.
Common filesystem is presently mounted read-only on:
        server_3
        server_2
Common filesystem is not mounted read-write anywhere.
```

Enable Unicode

Enabling Unicode on a datamover ensures that all filesystems created in the future will be encoded with Unicode, and therefore will not require encoding conversion for NFSv4 support.

Note: Please do not confuse Unicode encoding conversion and Access Policy conversion. These are two different processes and are independent of each other.

To enable Unicode on a datamover, use the following procedure:

```
$ /nas/sbin/uc_config -setup
$ /nas/sbin/uc_config -update
$ /nas/sbin/uc_config -on -mover <movername>
where:
<movername> = name of the Data Mover
```

Example:

Create a translation container:

```
$ /nas/sbin/uc_config -setup
Creating the common Unicode translation subdirectory.
Done
```

Update translation tables (copy from /nas/site/locate to DM:/.etc_common/xlt):

```
$ /nas/sbin/uc_config -update
Done
```

Enabled on datamover 2:

```
$ /nas/sbin/uc_config -on -mover server_2
server_2 : done
```

Check the status (alternative way)

```
$ server_cifs server_2 | grep I18N

Output:
I18N mode = UNICODE   <== if this line says ASCII Unicode is not on!
```

Convert file system to Unicode

In rare cases, where Unicode was enabled after filesystem creation, a filesystem must be manually converted to I18N.

Keep the following guidelines in mind:

- For a relatively small number of files and directories, the conversion should not take long and you should not notice any impact to system performance.
- When converting a large number of file systems and directories, you might experience some delay as file or directory names are converted.
- If you have several very large directories, you might want to perform the conversion during a period of low network activity.
- You can convert directories or files to Unicode from one local character encoding only. The desired character translation file is configured in the

uc_config -convert command, for example 8859-1.txt (which is a default US ASCII character set)

- If your environment supports more than one character encoding per Data Mover, contact EMC Customer Support for assistance in converting your system.

Note: The full list of I18N encoding sets can be found in /nas/site/locale for example, big5.txt that defines BIG5 Chinese character set.

Procedure

EMC recommends leaving Unicode conversion enabled to allow users to access (and convert) the majority of files or directories. The length of time this takes depends on the size of your file system and directories.

To start conversion, use the following command syntax:

```
$ /nas/sbin/uc_config -convert start <filename> -mover <movername>
where:
<filename> = name of the character translation file that should be used to
convert data to Unicode.

<movername> = name of the Data Mover for which you want to start Unicode
conversion. This option causes Unicode conversion to start for all file
systems on the specified Data Mover. If ALL is used, Unicode conversion
starts for all Data Movers in VNX cabinet.
```

Example:

To start conversion of data from US ISO 8859-1 character set to Unicode for all file systems on server_3, type:

```
$ /nas/sbin/uc_config -convert start 8859-1.txt -mover server_3
Output:
You are using the translation file (8859-1.txt), which will translate ISO
8859-1 (1987) to Unicode.

It is recommended that you back up your file system before running this
command. Have you backed up your file system?

Please make sure that you are using the right translation file, otherwise
data on the disk will be corrupted and unrecoverable!

Do you wish to continue [yes or no] Operation cannot be interrupted.
yes
```

Find and convert files

You can find and convert files at any time after enabling Unicode. Running find or search commands immediately after starting the Unicode conversion causes each file or directory name to be enumerated and converted without waiting for users to specifically access them. This is recommended if you want to convert everything at once.

To convert files gradually, run the find or search commands after most or all of the directories have been accessed. The purpose of running the find or search commands is to catch a small number of directories or files that have not been accessed yet. Determine the best timing for your system before deciding when to run these commands.

Type the find command from any UNIX or Linux client, or perform a search in a Windows environment so that every directory on each effected Data Mover is found. This enumerates and converts to Unicode files in each directory (if not already done).

For example:

```
$ find <mountpath> - name randomNonexistentFile
```

Stop Unicode conversion

To stop Unicode conversion after converting all files and directories, use the following command syntax:

```
$ /nas/sbin/uc_config convert stop <filename> mover <movername>|ALL}

where:
<filename> = name of the character translation file that is being used to
convert data to Unicode.

<movername> = name of the Data Mover for which you want to stop Unicode
conversion. This option causes Unicode conversion to stop for all file
systems on the specified Data Mover. If ALL is used, Unicode conversion
stops for all Data Movers in VNX cabinet.
```

Example:

To stop Unicode conversion (8859-1 encoding) for all file systems on a specified Data Mover, type:

```
$ /nas/sbin/uc_config -convert stop 8859-1.txt -mover <movername>

where:
<movername> = name of Data Mover
```

Rerun the conversion

When you run the uc_config -convert start command, VNX creates a timestamp. If the timestamp of a directory is older than the conversion timestamp, VNX converts the files in the directory. After conversion, the timestamp of the directory is later than the timestamp of the directory before conversion. The next user access of the directory does not trigger conversion.

However, if you need to run the uc_config -convert start command again, for example in a case where the NFS client has the wrong locale, you can simply reissue the uc_config –convert start command to touch all the directories again. Only unconverted files are updated. Files that are already converted are skipped.

NFSv4 ACLs

User access control by using ACLs

NFSv4 adds support for access control lists (ACLs). The ACLs provide finer-grained user access control to file system objects than traditional NFS mode bits. The VNX has a single implementation of ACLs that is based on a Microsoft Windows ACL. As such, VNX NFSv4 ACLs are not POSIX compliant[1].

The VNX performs ACL-based access control for NFS clients when the file system is mounted with one of the following VNX access policies: NT, SECURE, MIXED, or MIXED_COMPAT.

To provide NFSv4 clients with the ability to manage file system object ACLs, mount the file system by specifying the access policy MIXED or MIXED_COMPAT. The access policy is specified by using the accesspolicy option on the server_mount command. The *Managing a Multiprotocol Environment on VNX* technical module *provides* detailed information about configuring the VNX access control policies.

Note: Although the NFSv4 standard defines ACL syntax that is similar but not equivalent to the Windows ACL syntax, the relationship of items such as Access Control Entry (ACE) order and number of ACEs for each principal is not defined. Therefore, Windows, UNIX, and Linux clients might manage file system object ACLs differently. In some cases, an ACL set by a Windows client might not be acceptable to a UNIX or Linux client. The client will fail to display the ACL. However, access control to the file system objects is not affected.

ACL tools

Since VNX NFSv4 ACL is not POSIX compliant, not all UNIX tools work with the VNX. The table below reviews some of the common toolsets.

Operating System	Tools set	VNX NFSv4 compatibility
Linux	Extended chmod and ls	Yes
Linux	Extended getfacl and setfacl	Yes
Linux	AFS package: fs setacl, fs listacl, etc.	No
SunOS/Solaris	ZFS package: getfacl and setfalc	No

[1] NFSv4 standard does not define command line interface, so POSIX compliance is outside its scope

SunOS/Solaris	Extended chmod and ls	Yes

Examples:

To list ACLs on a file

```
$ ls -v file.1
Output:
-rw-r--r-- 1 root root 2703 Nov 4 12:37 file.1
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_attributes/
       write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/write_attributes/
       write_acl/write_owner:deny
5:everyone@:read_data/read_attributes/read_acl/synchronize:allow
```

To add an ACL to a file

```
$ chmod A+user:nasadmin:read_data:allow file.1
$ ls -v file1
Output:
-rw-r--r-- 1 root root 2703 Nov 4 12:37 file.1
0:user:nasadmin:read_data:allow
1:owner@:execute:deny
2:owner@:read_data/write_data/append_data/write_attributes/
       write_acl/write_owner:allow
3:group@:write_data/append_data/execute:deny
4:group@:read_data:allow
5:everyone@:write_data/append_data/write_xattr/write_attributes/
       write_acl/write_owner:deny
6:everyone@:read_data/read_attributes/read_acl/synchronize:allow
```

Replace ACL

```
$ chmod A0=user:nasadmin:read_data/write_data:allow file.1
$ ls -v file1
Output:
-rw-r--r-- 1 root root 2703 Nov 4 12:37 file.1
0:user:nasadmin:read_data/write_data:allow
1:owner@:execute:deny
2:owner@:read_data/write_data/append_data/write_attributes/
       write_acl/write_owner:allow
3:group@:write_data/append_data/execute:deny
4:group@:read_data:allow
5:everyone@:write_data/append_data/write_xattr/write_attributes/
       write_acl/write_owner:deny
6:everyone@:read_data/read_attributes/read_acl/synchronize:allow
```

Troubleshooting NFSv4

When you encounter problems while using NFSv4, review the following:

- The config file and verify that the hivers option is set to 4.

- The server_mount command and verify that the accesspolicy option is set to MIXED or MIXED_COMPAT.
- NFSv4 domain parameter:
  - Must be set. Otherwise, users and groups are mapped to nobody.
  - Server and clients use the same NFSv4 domain name.
- NFSv4 client system's mount command and verify that it specifies version 4.

You can also validate:

- Connectivity:
  - Ping from the Data Mover to the Key Distribution Center.
  - Ping from the Data Mover to the client.
- If using NIS:
  - Ping from the Data Mover to the NIS server.
  - Verify the users or groups having problems are in the NIS passwd, group, or gsscred_db files.
- Naming service configuration (NFSv4 server and clients must access the same information).

Display NFSv4 status

To display the status of the NFSv4 service, use this command syntax:

```
$ server_nfs <movername> -v4

where:
<movername> = name of the Data Mover
```

Example:

To display the status of the NFSv4 service on server_2, type:

```
$ server_nfs server_2 -v4
Output:
server_2:
---------- nfsv4 server status ----------
* service started *
----- nfsv4 clients -----
configured clients: 5
unconfirmed clients: 0
-------------------------
----- nfsv4 state -----
opens: 8
locks: 4
delegations: 0
```

Display NFSv4 statistics by using server_stats command

To display NFSv4 statistics, use the following command:

```
$ server_stats server_2 -m nfs.v4
Output:
server_2 NFS V4IO    Read    Write NFS Op     NFS      NFS       NFS
Timestamp            Cnts/s Cnts/s            Op       Op         Op
                                              Calls/s uSec/Call  %
11:02:04 32768-65535   0    35593 v4Compound 36121   2801       25
                                  v4Commit     554      3        0
                                  v4GetAttr  36111      3       25
                                  v4PutFh    36151      5       25
                                  v4Write    35559   2277       25
11:02:05 32768-65535    0    33795 v4Compound 34365   4397       25
                                  v4Commit     530      3        0
                                  v4GetAttr  34354      3       25
                                  v4PutFh    34323      5       25
                                  v4Write    33822   3937       25
.
.
.
```

Verify the VNX-side setup

To review the Data Mover logs and or configuration see the following commands followed by examples.

server_log: For any issues with authentication of DNS or NFSv4 errors use the following syntax to output the last 50 lines of the server log:

```
$ server_log <movername> –a –s |tail 50
```

```
where:
<movername> = the name of the Data Mover
```

server_ldap: To verify the LDAP setup if applicable, use the following syntax:

```
$ server_ldap <movername> -info
```

```
where:
<movername> = name of the Data Mover
```

server_nis: To verify the NIS setup if applicable, use the following syntax:

```
$ server_nis <movername>
```

```
where:
<movername> = name of Data Mover
```

server_mount: To verify that the file system is mounted and the correct options are set, use the following syntax:

```
$ sever_mount <movername>
```

```
where:
<movername> = name of Data Mover
```

server_export: To verify that the file system is exported with the required options, use the following syntax:

```
$ server_export <movername>

where:
<movername> = name of Data Mover
```

Check Data Mover files

You can verify the following files by using the server_file command on the Data Mover:

- nsswitch.conf
- hosts (if applicable)
- passwd
- group

Example:

```
$ server_file <movername> -get <filename> <filename>

where:
<movername> = name of the Data Mover
       <filename> = name of the file you want to retrieve and the second
 <filename> is what you want it to be retrieved as.
```

The server_file command retrieves the file from the Data Mover where you can review the correct information on the working directory

Check the client-side logs

Refer to the operating-system-specific documentation for more information on the host logs.

Operating System	Location
SunOS	/var/adm/messages
Linux	/var/log/messages
AIX	man pages for errpt and alog
HP-UX	/var/adm/syslog/syslog.log

Appendix A: Set up client OS's to use with NFSv4

For most operating systems, you can use the showmount command to show remote NFS mounts (resources).

To mount an NFS file system, the resource must be available on the NAS or NFS server. To verify that resource is available, open the terminal and type the following command:

```
$ showmount -e <movername>

Where:
<movername> = name of the Data Mover
```

Linux (RHEL and/or SUSE)

Required services

RHEL uses a combination of kernel-level support and daemon processes to provide NFS file sharing. All NFS versions rely on remote procedure calls (RPC) between clients and servers. RPC services under Red Hat Enterprise Linux 6 are controlled by the rpcbind service. To share or mount NFS file systems, the following services work together, depending on which version of NFS is implemented.

Note: The portmap service was used to map RPC program numbers to IP address port number combinations in earlier versions of Red Hat Enterprise Linux. This service is now replaced by rpcbind in Red Hat Enterprise Linux 6 to enable IPv6 support.

NFSv4 Domain name

First, make sure idmapd daemon is enabled in /etc/default/nfs-common, and set NEED_IDMAPD=yes.

Then, set the NFSv4 domain in /etc/idmapd.conf file, by configuring domain=NFSV4_domain there.

NFS client configuration

The mount command mounts the NFS shares on the client side. The format is as follows:

```
# mount -t nfs -o options host:/remote/export /local/directory
```

The NFS protocol version used in Red Hat Enterprise Linux 6 is identified by the mount options nfsvers or vers. By default, mount uses NFSv4 with mount -t nfs. If the server does not support NFSv4, the client automatically steps down to a version supported by the server. If you use the nfsvers/vers option to pass a particular version not supported by the server, the mount will fail. The file

system type nfs4 is also available for legacy reasons; this is equivalent to running

```
# mount -t nfs -o nfsvers=4 host:/remote/export /local/directory
```

Refer to man page of the mount command for more details.

If an NFS share is mounted manually, the share will not be automatically mounted upon reboot. Red Hat Enterprise Linux offers two methods for mounting remote file systems automatically at boot time: the /etc/fstab file and the autofs service.

Mount

An alternate way to mount an NFS share from another machine is to add a line to the /etc/fstab file. The line must specify the hostname of the NFS server, the directory on the server being exported, and the directory on the local machine where the NFS share is to be mounted. You must be root to modify the /etc/fstab file.

The general syntax for the line in /etc/fstab is as follows:

```
server:/usr/local/pub /pub nfs defaults 0 0
```

The mount point /pub must exist on the client machine before this command can be executed. After adding this line to /etc/fstab on the client system, use the command mount /pub, and the mount point /pub is mounted from the server.

The /etc/fstab file is referenced by the netfs service at boot time, so lines referencing NFS shares have the same effect as manually typing the mount command during the boot process.

A valid /etc/fstab entry to mount an NFS export should contain the following information:

```
server:/remote/export /local/directory nfs options 0 0
```

The variables server, /remote/export, /local/directory, and options are the same ones used when manually mounting an NFS share.

Note: The mount point /local/directory must exist on the client before /etc/fstab is read. Otherwise, the mount will fail. For more information about /etc/fstab, refer to man fstab.

Automount with NFSv4

To automount an NFSv4 exported volume using autofs.

Two files that allow automount to work using autofs reside under the /etc directory:

- `Auto.master`
- auto.misc `or` auto.home `or` auto.xxxxxx`. Here` xxxxxx `can be any name.`

auto.master file contains

```
#
# $Id: auto.master,v 1.4 2005/01/04 14:36:54 raven Exp $
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#/misc  /etc/auto.misc --timeout=60
#/smb   /etc/auto.smb
#/misc  /etc/auto.misc
#/net   /etc/auto.net

/export /etc/auto.misc
```

In the above file, the auto.misc file can also be auto.home or auto.*xxxxxxx* . The corresponding entry in the auto.misc or auto.home or auto.*xxxxxx* should be the one below. (Here auto.misc is used.)

```
#
# $Id: auto.misc,v 1.2 2003/09/29 08:22:35 raven Exp $
#
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

cd         -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
export     -fstype=nfs4,rw     NFSServer:/

# the following entries are samples to pique your imagination
#linux           - ro,soft,intr        ftp.example.org:/pub/linux
#boot            -fstype=ext2          :/dev/hda1
#floppy          -fstype=auto          :/dev/fd0
#floppy          -fstype=ext2          :/dev/fd0
#e2floppy        -fstype=ext2          :/dev/fd0
#jaz             -fstype=ext2          :/dev/sdc1
#removable       -fstype=ext2          :/dev/hdd
```

After making these entries restart autofs:

```
# /etc/init.d/autofs restart
```

To check the status of the autofs, type:

```
# /etc/init.d/autofs status
```

Automount mounts the path as soon as it is addressed, for example, even a simple list command will mount it:

```
 $ ls /mntpoint/exportedDir
```

After executing the above command, visually confirm the successful mount:

```
# df -h
Filesystem    1K-blocks Used    Available  Use%  Mounted on
/dev/hda1     31462264  2810852 28651412   9%    /
udev          518296    88      518208     1%    /dev
Nfsserver:/   69575360  2268576 67306784   4%    /export/export
```

Using /etc/fstab to mount NFSv4 exported volume

The NFS exported volume can also be mounted on the client just by making an entry in the /etc/fstab file. If your NFS server name is NFSserver and the mount point on the client is /mnt_point then the entry in the fstab should look like something below.

```
/dev/sda1          /              reiserfs defaults          1  1
/dev/sda2          swap           swap     defaults          0  0
proc               /proc          proc     defaults          0  0
sysfs              /sys           sysfs    noauto            0  0
usbfs              /proc/bus/usb  usbfs    noauto            0  0
devpts             /dev/pts       devpts   mode=0620,gid=5   0  0
/dev/fd0           /media/floppy  auto     noauto,user,sync  0  0

NFSserver:/export  /mnt_point     nfs4     rw,user,noauto    0  0
```

After making this entry in the /etc/fstab file, at the command prompt of the client just give the command: mount /mnt_point.

To check the mount points on the client, type mount.

```
$ mount
/dev/hda3 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
nfsd on /proc/fs/nfsd type nfsd (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)

NFSserver:/export on /mnt_point type nfs4 (rw,addr=xxx.xxx.xxx.xxx)
```

Here xxx.xxx.xxx.xxx is the IP address of the NFS server.

You can check the highlighted line to find out which version of NFS mount is done.

Solaris 10

Ensure the NFS utilities package is present on the system.

Note this may install dependencies.

```
# yum install -y nfs-utils nfs4-acl-tools
[. . .]
```

Configure NFSv4 domain

In /etc/default/nfs file, configure NFSMAPID_DOMAIN=NFSV4_domain_name

Mount

```
# mount -o  vers=4,soft,intr,nolock,wsize=32768,rsize=32768
```

AIX

Configure NFSv4 Domain

The NFSv4 domain is configured using chnfsdom command

```
# chnfsdom -a nfsv4_domain nfsv4_domain
```

Mount

After the domain is set, it is time to mount:

```
# mount -o vers=4 vnx_nfs4_server:/path/to/export /mnt
```

HP-UX

Configure NFSv4 Domain

Edit /etc/default/nfs file and set two variables:

- `NFS_CLIENT_VERSMAX=4`
- `NFSMAPID_DOMAIN=NFSV4-domain-string`

Mount

After the domain is set, it is time to mount:

```
# mount -o overs=4 vnx_nfs4_server:/path/to/export /mnt
```