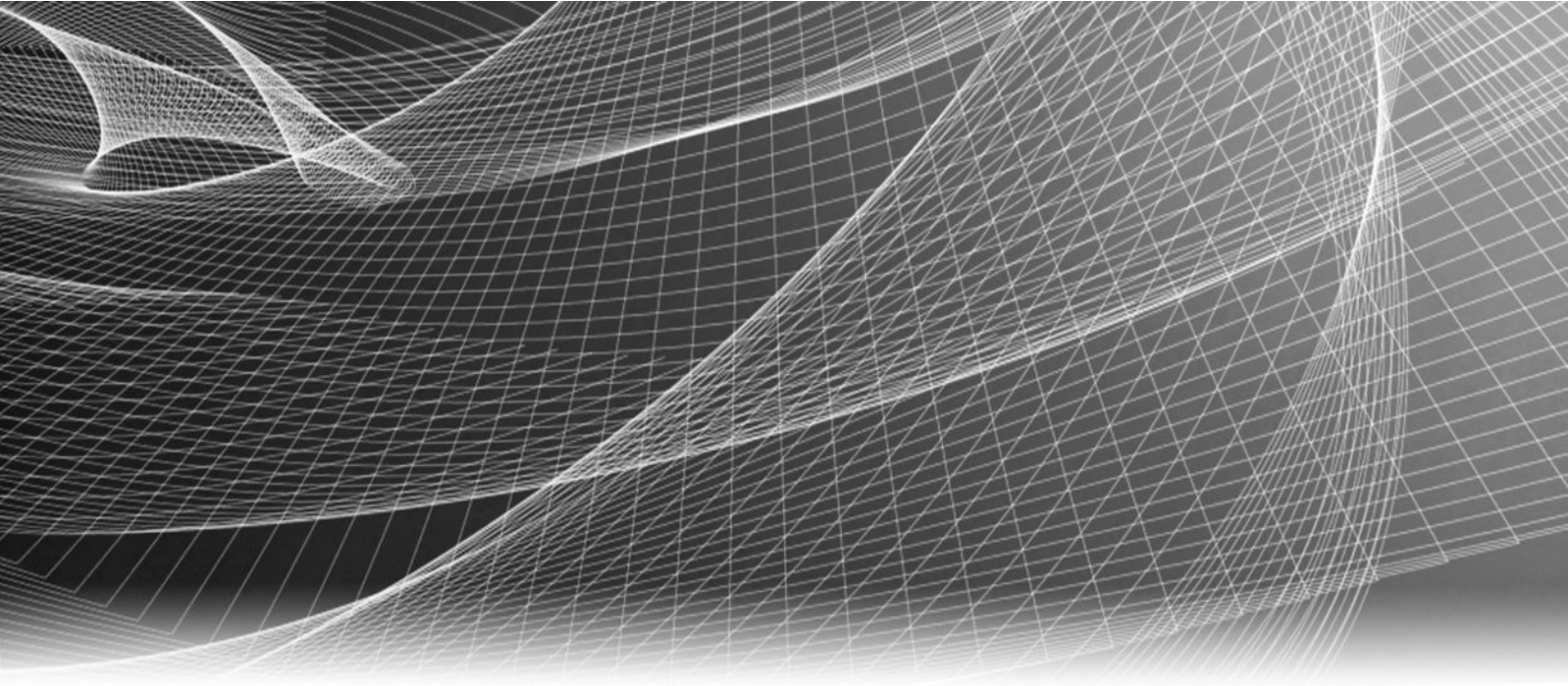


EMC® VNX® Series

Version 8.1

Configuring NFS on VNX

P/N 300-014-336 REV. 02

EMC²

Copyright © 1998-2014 EMC Corporation . All rights reserved. Published in USA.

Published July, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com).

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

CONTENTS

Preface 7

Chapter 1 Introduction 9

System requirements 10
Restrictions 10
Cautions 12
User interface choices 13
Related information 13

Chapter 2 Concepts 15

Overview 16
VDM for NFS Multiple Domains Solution 16
Secure NFS for VDM 18
Planning considerations 18
System access control 18
User authentication 19
User access control by using mode bits 20
User access control by using ACLs 21
Internationalization 21
NFSv4 domain name 21
NFSv4 file delegation 22
File locking 22
Client contexts 24
Limiting access to NFSv4 clients only 24

Chapter 3 Configuring 25

Getting started 26
Enable NFSv4 26
Configure the NFSv4 domain 26
Create a user file system 28
Configure access to naming services 28
Verify or Enable Unicode 29
Create a mount point 30
List the mount point 31
Mount a file system 31
Mount a file system for basic NFS access 31
Mount a file system for NFSv4 access 32
Unmount a file system 33
Export a file system 33
Export a file system for basic NFS access 34
Levels of access 35
Prevent setuid and setgid bit access 36
Multiple levels of a file system 37
Restrict access by VLAN 38
Specify NFSv4 access 39
Specify the user authentication method 39
Mount a NFS share on MAC OS using the GUI 41

Mount a NFS share on MAC OS manually 42
Configure secure NFS by using a Windows Kerberos KDC 42
Enable dynamic updates of reverse lookups 43
Configure CIFS 43
Set the secure NFS service instance 46
Set the NFS user and group mapping service 48
Reorder ACEs (multiprotocol environment only) 49
Configure secure NFS by using a UNIX or Linux Kerberos KDC 50
Set the Data Mover name 51
Configure the Kerberos realm 51
Set the secure NFS service instance 53
Create the NFS Kerberos service principals 55
Map user principal names to UIDs 57
Use the Solaris mapping utility gsscred 59
Copy the mapping file 59
Copy the mapping file to an NIS server 60
Copy the mapping file to a Data Mover 61
Create a local mapping file 61
Mount a VNX file system on a client system 63

Chapter 4 Managing 65

Manage NFS 66
Unexport the NFS path to a VNX file system 66
Reexport all NFS paths on a VNX 67
Disable NFS access to all file systems on a Data Mover 67
NFS automounter feature support 68
Hide NFS exports 70
Manage NFSv4 72
Modify state duration 72
Change the delegation recall timeout 72
Modify number of usable nodes 73
Display NFSv4 service status 73
Stop the NFSv4 service 74
Restart the NFSv4 service 74
Display NFSv4 clients 75
Display information about NFSv4 clients 75
Release NFSv4 clients 76
Support 32-bit and 64-bit NFS clients 77
Manage secure NFS 77
View keytab entries 78
Display all secure NFS service instances 78
Display user attributes 79
Display information about a user in a local mapping file 79
Delete a user from a local mapping file 80
Delete service principals 80
Release authentication 80

Chapter 5 Troubleshooting 83

EMC E-Lab Interoperability Navigator 84
VNX user customized documentation 84
Troubleshooting NFSv4 84
Display NFSv4 status 84
Display NFS statistics 85
Troubleshooting secure NFS 86

Cannot access file system 87
Release user authentication 88
Error messages 88
NFSv4 error messages 88
EMC Training and Professional Services 89

Appendix A System Access Behavior 91

Behavior when specifying combinations of access modes 92
Rules for resolving conflicts among hosts, subnets, and netgroups 92
Enable NFS special look up order for netgroups 95
Specify read-only as the default access mode 95

Appendix B User Authentication Behavior 97

General rules when specifying security options 98
Root access mode 98

Appendix C NFS Authentication Daemon for PC Clients 101

PC client access 102
Set up PC client software 102
Hummingbird PC NFS client issues 103

Appendix D Parallel NFS on VNX for Block 105

About pNFS 106
Configuring pNFS for server 107
Set up pNFS service 107
Stop pNFS 108
Restart pNFS 108
Mount the file system for the pNFS service 109
Export a file system for pNFS access 109
Set up pNFS client 109
Configuring SAN and storage 110
Install the Fibre Channel switch (FC configuration) 110
Zone the SAN switch (FC configuration) 111
Configure the iSCSI-to-Fibre-Channel bridge (iSCSI configuration) 111
Enable iSCSI on the MDS console (iSCSI configuration) 112
Configure the MDS iSCSI port for Gigabit Ethernet (iSCSI configuration) 112
Configure the iSCSI port proxy and TCP parameters (iSCSI configuration) 112
Create and configure a new zone for iSCSI proxy (iSCSI configuration) 113
Activate the zone (iSCSI configuration) 113
Enable SP A and SP B FC interfaces (FC configuration) 114
Add a VSAN for iSCSI/FC and enabling zones 114
Create a security file on a VNX for File 114
VNX for block iSCSI port configuration 115
Access Logix configuration 115

Index 117

Preface

As part of an effort to improve and enhance the performance and capabilities of its product lines, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

If a product does not function properly or does not function as described in this document, please contact your EMC representative.

Special notice conventions used in this document

EMC uses the following conventions for special notices:

DANGER

Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

Addresses practices not related to personal injury.

Note

Presents information that is important, but not hazard-related.

Where to get help

EMC support, product, and licensing information can be obtained as follows:

Product information—For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to EMC Online Support (registration required) at http://Support.EMC.com.

Troubleshooting—Go to EMC Online Support at http://Support.EMC.com. After logging in, locate the applicable Support by Product page.

Technical support—For technical support and service requests, go to EMC Customer Service on EMC Online Support at http://Support.EMC.com. After logging in, locate the applicable Support by Product page, and choose either **Live Chat** or **Create a service request**. To open a service request through EMC Online Support, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

Note

Do not request a specific support representative unless one has already been assigned to your particular system problem.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications.

Please send your opinion of this document to:

techpubcomments@EMC.com

CHAPTER 1

Introduction

The EMC VNX provides access to data through a variety of file access protocols, including the Network File System (NFS) protocol. NFS is a client or server distributed file service that provides file sharing in network environments. When configured as an NFS server, VNX provides access to VNX file systems for clients that use versions 2, 3, and 4 of the NFS protocol.

This document is part of the VNX information set and is intended for use by system administrators responsible for configuring and managing NFS on the system.

Topics included are:

- System requirements 10
- Restrictions 10
- Cautions 12
- User interface choices 13
- Related information 13

System requirements

System requirements on page 10 describes the EMC® VNX® software, hardware, network, and storage configurations required for configuring NFS as described in this document.

Table 1 System requirements

Software	VNX OE for File 8.1 For secure NFS that uses UNIX or Linux-based Kerberos: Sun Enterprise Authentication Mechanism (SEAM) software or Linux key distribution center (KDC) that runs Kerberos version 5. **Note** KDCs from other UNIX systems have not been tested. For secure NFS that uses Windows-based Kerberos: Windows 2000 or later. For secure NFS, the client computer must be running: • SunOS version 5.8 or later (Solaris 10 for NFSv4). • Linux-kernel 2.4 or later (2.6.12.‹X› NFSv4 patches for NFSv4). • Hummingbird Maestro version 7 or later (EMC recommends version 8); version 9 for NFSv4. • AIX 5.3 ML3. **Note** Other clients have not been tested. • DNS. • Network Time Protocol (NTP) server. Windows environments require that you configure VNX in the Active Directory. For Parallel NFS (pNFS), ensure that the Linux kernel that you use supports pNFS block. Currently, only Fedora has a release supporting pNFS block.
Hardware	No specific requirements.
Network	If IPv6 is being deployed, configure both IPv4 and IPv6 addresses for DNS, NIS, and NTP servers. This way if an IPv6-only situation arises for any reason, then these services will still work. If a Data Mover with only IPv6 addresses is desired, it is imperative that all these services have IPv6 addresses configured in order to work.
Storage	No specific requirements.

Restrictions

The VNX supports all NFS features with the following restrictions:

- Secure NFS is supported only for NFSv3 and NFSv4.

- If you are using the VNX FileMover feature or CDMS, the NFS communication between the VNX and secondary storage requires UNIX AUTH_SYS authentication and does not support NFSv4.
- The NFS endpoint on a VDM works for TCP only (NFSv3 and NFSv4). For NFSv2 and NFSv3 over UDP, you cannot use a VDM.
- You can perform the NFS export of a file system mounted on the VDM by using the CLI `server_export` command and not by using EMC Unisphere® software.
- Secure NFS is not supported on a VDM NFS endpoint. Like for UDP, secure NFS works only for the endpoint on the physical Data Mover.
- NFS export of a file system mounted on VDM can be done through the CLI server_export and not with EMC Unisphere software.
- NFS export cannot be restricted to a specific VDM. However, the virtual local area network (VLAN) option in server_export can be used.
- Data on a VDM mounted file system can be accessed from an NFS client by using celerra:/root_vdm_x/fs_name where x is a consecutive number assigned by VNX when you create the VDM. The NFS export aliasing can be used to change the view of NFS exports to the client.

The VNX implementation of NFSv4 supports:

- NFSv4 over the networking protocol IPv4
- UNIX and Kerberos v5 for user authentication
- UTF-8 and ASCII
- Pseudo-root file systems
- Access control lists
- Nested mount file systems

The VNX implementation of NFSv4 does not support:

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

The VNX implementation of pNFS does not support the following type of file systems:

- EMC SnapSure™ checkpoints.
- Replicated copies of the file systems.
- FLR-C File Systems.
- File systems that are unaligned at the volume layer.
- File systems that are undergoing a state transition. For example, file systems restoring from a checkpoint.

The file systems unaligned at the volume layer can occasionally be created directly on disk volumes. Therefore, it is highly recommended that all file systems intended for use by pNFS be created upon a slice volume.

VDM replication restrictions

- A file system replica for an NFS access should have the same file system ID as the source:
  - The NFS file handle exchanged across the wire includes the file system identifier.
  - The nas_replicate -pool option makes the best effort to satisfy this requirement, unless the source file system ID conflicts with an existing file system on the destination.
- You can replicate a file system without having the same FSID. However, upon failover the clients will have to remount the Data Mover exports.

Replicating a VDM for NFS with ESX datastore

When replicating a VDM for NFS file systems that are used for ESX datastores you must follow the following procedure to ensure that the VMs do not go offline.

Note

ESX server will lose access to the datastore if these steps are not followed exactly in the given order.

This procedure assumes that you have your NFS data store connecting to the VNX through an IP address.

- Failover the Production File System that is mounted on the VDM
- Down the Interface on the Source
- Failover the VDM
- Up the Interface on the Destination

⚠ CAUTION

If the VDM is failed over before the user file system, the ESX server receives an error NFS3ERR_STALE(Invalid File Handle). The ESX client then considers the file system to be down. To avoid this error, when the Production File System (PFS) that is mounted on a VDM is failed over, the VDM interface is set down using the server_ifconfig command on the source. The VDM is then failed over. As the VDM restarts on the replication site, the VDM interface can be set up using the server_ifconfig command.

Cautions

If any of this information is unclear, contact your EMC Customer Support Representative for assistance.

Security

NFSv4.1 does not introduce any security issues beyond those in NFSv4.0. The server listens on a well-known port (2049) and does not use portmapper to assign ports.

NFSv4.1 does address two inadequacies inherent in NFSv4.0: denial-of-service attacks through unauthorized changes to client state and firewalls' blocking of callbacks. pNFS adds data security risks.

Session security

For NFSv4.1 clients that uses RPCSEC_GSS based security mechanisms, which is offered on VNX and must be implemented by any compliant NFSv4.0 server. The issue is solved by using a digest to protect the binding of connections to sessions. The digest is computed from a shared secret by using a hash function.

The level of session security and the hash algorithm are negotiated during session creation. The client sets a shared secret called a secret session verifier or SSV with the

SET_SSV call. This call should be made immediately after session creation before binding any connections to the session. When a connection is bound to the session, the arguments to BIND_CONN_TO_SESSION include the digest computed from the client's secret. This makes it all but impossible for an attacker to bind its own connection to the session and disrupt it with spurious calls.

Firewall and callbacks

In NFSv4.0, the callback may fail if the firewall prevents the server from opening the port connection. The NFSv4.1 protocol avoids the issue by binding the connections opened by the client to the back channel of the session.

Data security

In pNFS with block/volume layouts, clients have parallel access to the storage devices. Permissions to write blocks to the storage devices are managed by the server by granting and revoking layouts, but an uncooperative client can still write to a layout after it has been returned, and it can write outside the boundaries of any layout it has received.

If client-side protection is insufficient, then pNFS should not be used.

User interface choices

The VNX offers flexibility in managing network storage that is based on the support environment and interface preferences. This document describes how to configure NFS by using the command line interface (CLI). You can also perform some of these tasks by using the EMC Unisphere software. From the navigation pane, select NFS Exports to export a path to the file system and specify the level of access for each export.

For additional information about managing your VNX:

- EMC VNX Documentation on EMC Online Support
- Unisphere online help

Installing VNX Management Applications includes instructions on launching Unisphere, and on installing the MMC snap-ins and the ADUC extensions.

The *VNX for File Release Notes* contain additional, late-breaking information about VNX management applications.

Limitations

You cannot use the Unisphere to configure the following features:

- Automounter
- Secure NFS
- NFSv4 features
- VDM for NFS multiple domains solution

Related information

For specific information related to the features and functionality described in this document:

- *VNX Glossary*
- *Configuring and Managing CIFS on VNX*
- *Managing VNX for a Multiprotocol Environment*
- *Using International Character Sets on VNX for File*
- *Configuring VNX Naming Services*

- *Configuring Time Services on VNX*
- *Managing Volumes and File Systems for VNX Manually*
- *EMC VNX Command Line Interface Reference for File*
- VNX for File man pages
- *Parameters Guide for VNX for File*

Other related publications include:

- [RFC1094] *Network File System Protocol Specification* (NFS Version 2 protocol specification), March 1989.
- [RFC1510] *The Kerberos Network Authentication Service (V5),* September 1993.
- [RFC1813] *NFS Version 3 Protocol Specification,* June 1995.
- [RFC1964] *The Kerberos Version 5 GSS–API mechanism,* June 1996.
- [RFC2203] *RPCSEC_GSS Protocol Specification,* September 1997.
- [RFC2623] *NFS Version 2 and Version 3 Security Issues and the NFS Protocol's use of RPCSEC_GSS and Kerberos V5,* June 1999.
- [RFC3530] *Network File System (NFS) Version 4 Protocol,* April 2003.

EMC VNX documentation on EMC Online Support

The complete set of EMC VNX series customer publications is available on EMC Online Support. To search for technical documentation, go to http://Support.EMC.com. After logging in to the website, click **Support by Product** and type **VNX series** in the Find a Product text box. Then search for the specific feature required.

VNX wizards

Unisphere software provides wizards for performing setup and configuration tasks. The Unisphere online help provides more details on the wizards.

CHAPTER 2

Concepts

The concepts and planning considerations to understand NFS are:

- Overview 16
- VDM for NFS Multiple Domains Solution 16
- Secure NFS for VDM 18
- Planning considerations 18

Overview

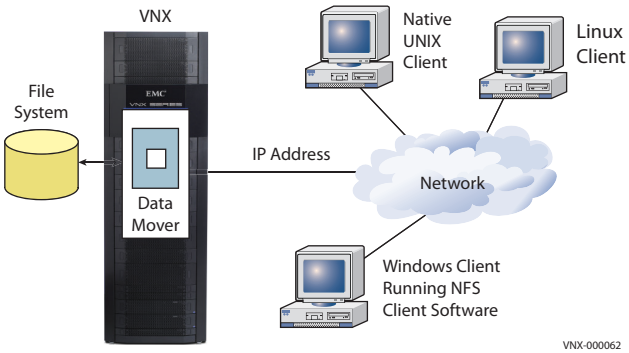
NFS environments can include:

- UNIX clients
- Linux clients
- Windows systems configured with third-party applications that provide NFS client services (for example, Hummingbird)

When a VNX for File is configured as an NFS server, file systems are mounted on a Data Mover and a path to that file system is exported. Exported file systems are then available across the network and can be mounted by remote users.

NFS file system configuration on page 16 shows a Data Mover configured for NFS clients.

Figure 1 NFS file system configuration



An NFS-configured Data Mover does the following:

- Provides access to the exported file system through an IP network.
- Authenticates the user if using a secure NFS.
- Performs authorization by:
  - Comparing the access rights of the NFS client by requesting information with the access rights defined for the exported file system.
  - Performing user access control on the file system object.

The VNX supports versions 2, 3, and 4 of the NFS protocol (NFSv2, NFSv3, and NFSv4). The NFS version 4 protocol is a further revision of the NFS protocol defined by versions 2 and 3. It retains the essential characteristics of previous versions (a design independent of transport protocols, operating systems, and file systems) and integrates file locking, strong security, operation coalescing, and delegation capabilities to enhance client performance.

Note

Managing a Multiprotocol Environment on VNX provides information about configuring the VNX to support both NFS and CIFS. *Configuring and Managing CIFS on VNX* provides information about CIFS only.

VDM for NFS Multiple Domains Solution

VNX provides a multinaming domain solution for the Data Mover in the UNIX environment by implementing a NFS server per Virtual Data Mover (VDM). The Data Mover hosting

several VDMs is able to serve the UNIX clients that are members of different LDAP or NIS domains, assuming that each VDM works for a unique naming domain. Similar to the CIFS service, several NFS servers are emulated on the Data Mover in order to serve the file system resources of the Data Mover for different naming domains. Each of these NFS servers is assigned to one or more different Data Mover network interfaces.

The following requirements are satisfied by this feature:

- The same Data Mover supports the NFS service for different LDAP, NIS, and DNS domains.
- Each domain is served by a different VDM addressed through a subset of network interfaces configured on the Data Mover.
- The file systems exported by a VDM can be accessed by CIFS, NFSv3, and NFSv4 over TCP protocols. The VDM solution classifies the file system resources. Consequently, only file systems mounted on a VDM can be exported by the VDM.
- Secure NFS, NFSv2, and NFSv3 over UDP protocols are not supported by this feature.
- The FTP, SFTP, and FTPS applications are not supported over a VDM. You can enable these applications on the physical Data Mover.

The VDMs loaded on a Data Mover use the network interfaces configured on the Data Mover. You cannot duplicate an IP address for two VDM interfaces configured on a same Data Mover. Once VDM interface is assigned, you can manage NFS exports on a VDM. The same network interface can be shared by both CIFS and NFS protocols. However, only one NFS endpoint and CIFS server is addressed through a particular logical network interface.

The multinaming domain solution implements an NFS server per VDM named 'NFS endpoint'. The VDM is used as a container that includes the file systems exported by the NFS endpoint and/or the CIFS server. These file systems of the VDM are visible through a subset of the Data Mover network interfaces attached to the VDM. The same network interface can be shared by both CIFS and NFS protocols on that VDM. The NFS endpoint and CIFS server are addressed through the network interfaces attached to that particular VDM.

Note

The multiple domain configuration is optional for a VDM. By default, the VDM functions for the users of the same domain as the Data Mover that hosts the VDM.

Note

You cannot use the Unisphere to configure this feature.

Configuring Virtual Data Movers on VNX provides more information about this feature.

VDM NFS exports

You can manage NFS exports per VDM by using the server_export command. All the NFS export options for VDM are supported.

The exports created on a VDM can export only file systems mounted on the VDM. The VDM NFS exports are automatically replicated as part of the VDM replication. The VDM NFS endpoint is automatically operational on destination site after the VDM switchover/failover. Moving a VDM from one Data Mover to another moves the whole NFS endpoint, including file systems and NFS exports.

Configuring Virtual Data Movers on VNX provides more information about managing NFS exports on a VDM.

Secure NFS for VDM

You can configure and manage secure NFS for VDM. You can use the `server_nfs` command and specify the VDM name as the mover name for the NFS protocol. The tasks to configure and manage the secure NFS service are the same as those for the physical Data Mover. Configure secure NFS by using a Windows Kerberos KDC on page 42 and Configure secure NFS by using a UNIX or Linux Kerberos KDC on page 50 provide more information about this.

The secure NFS service is started by default when a VDM is created and whenever the Data Mover is rebooted. However, if the service has been disabled through the CLI, it does not restart after the reboot.

Stopping the secure NFS service will do the following:

- Existing client secure connections will be terminated on the next incoming request or when they idle timeout, whichever is sooner.
- No new secure client connections will be allowed.
- If the secure NFS service is stopped and restarted, existing client secure connections will not resume, but will be terminated on the next incoming request or when they idle timeout, whichever is sooner.
- If the secure NFS service is stopped and the Data Mover is rebooted or the VDM is reloaded after a move or replication failover, the service will not be (re)started.

EMC Command Line Interface Reference for File describes `-secnfs` options for the server_nfs command.

Planning considerations

Based on the NFS environment requirements, review the following considerations before configuring NFS on the system:

- System access control on page 18
- User authentication on page 19
- User access control using mode bits on page 20
- User access control by using ACLs on page 21
- Internationalization on page 21
- NTFSv4 domain name on page 21
- NTFSv4 file delegation on page 22
- File locking on page 22
- Client contexts on page 24
- Limiting access to NTFSv4 clients only on page 24

System access control

To set the access rights of an NFS client system, use the server_export command. Clients authorized to access the export are specified through their hostname, netgroup, subnet, or IP address. A client system may have read-write access while a particular user might have only read access to a specific file. Export a file system for basic NFS access on page 34 describes this procedure. System access behavior on page 91 describes how to interpret the read and write access that you set for NFS clients of exported file systems.

User authentication

User authentication enables a system to validate the identity of a user, service, or server that is trying to access a resource or object, such as a file or a service. The VNX NFS service authenticates users through the mechanisms listed in User authentication methods on page 19.

Table 2 User authentication methods

Authentication method	Description
UNIX security	Serves as the default security. Using the AUTH_SYS security model, authentication of NFS users is assumed to be performed by the NFS client machine. The UID and GIDs are carried by the RPC protocol, which imposes a limit of 16 groups to which the user can belong. A new parameter security.maxgroups allows you to increase the number of UNIX groups for NFS clients from the default limit of 16 to 128. The *Parameters Guide for VNX for File* provides more information.
Secure NFS	Provides Kerberos-based user and data authentication, data integrity, and data privacy. Kerberos, a distributed authentication service, is designed to provide strong authentication that uses secret-key cryptography. When configured to act as a secure NFS server, the Data Mover uses the RPCSEC_GSS security framework and Kerberos authentication protocol to verify users and services. In a secure NFS environment, user access to NFS exported file systems is granted based on Kerberos principal names. However, access control to files and directories within the exported file system is still based on UID, GID, or ACLs. The Kerberos principal of the user is mapped to the UID by using a mapping service. **Note** Secure NFS on VNX supports more then 16 groups. Parameter security.maxgroups allows you to increase the number of UNIX groups for NFS clients from the default limit of 16 to 128.
Authentication daemon	For a Windows system (PC client) that uses NFS to access the VNX, an authentication daemon, typically rpc.pcnfsd or pcnfsd, is used to bridge the differences between Windows and UNIX NFS user authentication methods.

The different NFS versions support the following user authentication methods:

- All NFS versions support UNIX security and the PC authentication daemon.
- NFS versions 3 and 4 support secure NFS by using either UNIX or Linux or Windows Kerberos KDC.

Specify the user authentication method on page 39 describes how to specify the user authentication method by using the sec option on the server_export command. User Authentication Behavior on page 97 describes how to interpret the user authentication that was set for NFS clients of exported file systems.

NFS Authentication Daemon for PC Clients on page 101 describes how to use the authentication daemon.

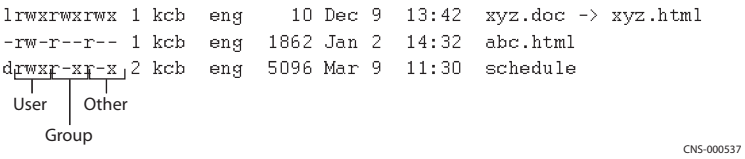
Configure secure NFS using a Windows Kerberos KDC on page 42 and Configure secure NFS using a UNIX or Linux Kerberos KDC on page 50 describe the procedures for configuring secure NFS by using either Windows or UNIX or Linux Kerberos.

User access control by using mode bits

The VNX performs mode-bit-based access control for NFS clients when the file system is mounted with one of the following system access policies: UNIX, NATIVE, SECURE, or MIXED_COMPAT. The access policy is specified by using the accesspolicy option on the server_mount command.

UNIX access rights are referred to as the mode bits of a file system object. They are represented by a string in which each bit represents an access mode or privilege granted to the user that owns the file, the owner group, and all other users.

UNIX mode bits are represented as three sets of concatenated rwx (read, write, and execute) triplets for each category of users (user or group or other), as shown in the following example of a file system directory:



In the example, the first 10 characters on each line indicate the file type and file permissions. The first character can be a d for a directory, a l for a symbolic link, or a dash (-) for a regular file.

The next three characters specify whether the user (kcb in the example) can read, write, or execute the file system object. The next three characters specify the permissions of the owner-group (owner primary group, eng in this example). The last three characters specify the permissions for all other users who are not the owner or members of the owner-group.

In the example, xyz.doc is a symbolic link that anyone can use to retrieve the xyz.html file. The abc.html is a regular file anyone can read, but only the user kcb can write to it. Anyone can search and read the schedule directory, but only the user kcb can insert files into it and delete files from it.

A UNIX credential is sent during the access request and consists of a user ID (UID) and up to 16 group IDs (GIDs) by default to which the user belongs. A new parameter security.maxgroups allows you to increase the number of UNIX groups for NFS clients from the default limit of 16 to 128. The *Parameters Guide for VNX for File* provides more information. When a user requests access to a file system object, the system compares the user's credentials with the permissions on that object.

The VDM for NFS multiple domain feature impacts the Windows-style credential for UNIX users. In order to support the NFS NTcredential domain for this feature, the following Windows registry entry is used to specify an equivalent parameter per VDM value. For a VDM, the 'nfs NTcred.winDomain' global parameter is replaced with the Windows registry key

```
[HKEY_LOCAL_MACHINE\Software\EMC\UnixNTCredential]
"NetbiosDomainName" = "<domain_name>"
```

When the VDM name resolution is confined to the VDM, the nfs NTcred feature uses the Windows domain specified in the VDM registry. Accordingly, a CIFS server member of this domain must be configured in the VDM and joined to the appropriate CIFS domain user to allow NFS access using a CIFS user credential.

Managing a Multiprotocol Environment on VNX provides more information about Windows-style credential for UNIX users.

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, which provides consistent access control no matter which protocol (NFS or CIFS) is used to access data.

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. Mount a file system for NFSv4 access on page 32 describes this procedure. *Managing a Multiprotocol Environment on VNX* 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 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.

Internationalization

The VNX supports clients in environments that use multibyte character sets. Multibyte character sets are supported by enabling universal character encoding standards (Unicode). The VNX also supports the UTF-8 specification for character encoding, as required by NFSv4. For UTF-8 clients, file and directory names are stored as received. The case of file and directory names are preserved, and no translation is required. For non UTF-8 clients, file system object names are translated to the UTF-8 specification.

Note

File systems that are not Unicode-enabled should be converted to Unicode before exporting them to NFSv4 clients. In addition, the VNX character encoding method must be changed from default to UTF-8.

If you are installing a new VNX, you can enable Unicode during the installation process before creating user files and directories. If you have an existing VNX, you must enable Unicode and then convert the files and directories. *Using International Character Sets on VNX for File* describes how to configure and manage Unicode.

NFSv4 domain name

Every VNX user must be identified by a unique numeric UID and GID to indicate the ownership of directories and files. The VNX uses directory and file ownership to apply and enforce access permissions and quota limits.

NFSv2 and NFSv3 use UIDs and GIDs to identify users and groups. Consequently, the system can use the UIDs and GIDs supplied by the NFS client to determine file ownership and access control. NFSv4, however, identifies users and groups by using a UTF-8

encoded user or group domain name, where the domain is the NFSv4 domain of which the server, clients, users, and groups are members. NFSv4 user and group domain names have the form user@domain or group@domain. These user and group domain names need to be converted to UIDs and GIDs.

Note

On the file system, the file owner and file owner-group are still represented by a numeric UID and GID regardless of NFS version.

Before you use NFSv4 on the VNX, you must specify an NFSv4 domain name by using the nfsv4.domain parameter. Configure the NFSv4 domain on page 26 describes how to specify the domain name.

Note

If the NFSv4 domain name is not specified on the server or on the NFSv4 clients, clients might be unable to access data. Ownership of files and directories is set to nobody/nobody.

NFSv4 file delegation

When using NFSv4, the VNX can delegate specific actions on a file to a client, such as specifically 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 you turn delegation off if:

- The data needs to be shared frequently by applications in 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.

Mount a file system for NFSv4 access on page 32 describes how to specify the delegation level. Manage NFSv4 on page 72 describes how to modify delegation operation by using NFSv4 parameters.

File locking

The VNX supports NFS file locking. NFSv4 file locking is similar to the Network Lock Manager (NLM) protocol used with NFSv2 and NFSv3, but is integrated into the NFSv4 protocol. This integration provides better support for different operating system semantics and error recovery.

Advisory and mandatory locking

In NFSv2 and NFSv3, locking rules are cooperative. A client is not prevented from accessing a file locked by another client if it does not use the lock procedure. VNX

considers NFSv2 and NFSv3 locks as advisory. An advisory lock does not affect read and write access to the file, but informs other users that the file is already in use.

NFSv4 provides mandatory locking, that is, the ability to block operations by other applications on a locked file. Mandatory locking is set by default for NFSV4, by using the parameter nfsv4.advisoryLocking (set to 1). You can switch to an advisory model for NFSV4.

Share reservation

A share reservation grants a process access to a file (read, write, both) and the ability to deny other processes access to the same file. For instance, an application can reserve a file for reading and writing, and deny writing by others.

NLM provides advisory share reservations for NFSv2 and NFSv3. An NLM share reservation does not prevent a noncooperative client from accessing a file.

NFSv4 supports mandatory share reservations with the OPEN operation. The share reservation is enforced for any file access.

Range locks

This lock applies to a range of bytes in a file. This range can cover the entire file. It is also referred to as a record lock.

Range locks can be of two types:

- Shared (read) locks
- Exclusive (write) locks

Several processes can hold a read lock on a particular file. But if a process holds an exclusive lock on a given byte range, no other process can hold any lock on this file for any range that includes part of the exclusively locked range until the exclusive lock on this range is removed. A read lock can be changed to an exclusive lock. Likewise, an exclusive lock can be changed to a read lock.

Lease

With NFSv2 and NFSv3, locks were granted by the server until released by the client. The drawback to this approach is that a crashed client keeps its locks until it recovers, if it recovers at all.

To avoid this drawback, NFSv4 locks are granted for a client lease duration only. Within the lease, the client is ensured that all its locks remain.

The lease is subject to renewal by the client. Any operation from the client renews its lease. If the lease expires, the server assumes that the client has failed. After a grace period, the server might allow other clients to acquire the same locks. If the server fails and restarts, it waits for at least the lease interval for clients to reclaim their locks before servicing any new lock request.

Mount a file system on page 31 describes how to configure locks on a file system based on the VNX.

STATD host status daemon

The STATD daemon monitors the NFSv2 and NFSv3 locks. One of the role of this daemon is to notify NFSv2 and NFSv3 clients that the NLM grace period starts when the Data Mover reboots. The notification messages sent to the clients include the STATD hostname.

For a VDM, the STATD hostname file is needed if you enable the VDM domain resolution with the `server_ns domains` command. The system will use the default Data Mover STATD hostname (specified in the STATD parameter) if you do not enable the VDM domain with the `server_nis` command. When you enable the VDM domain configuration, the hostname file must be created.

Parameters Guide for VNX for File provides more information about the STATD hostname parameter.

Client contexts

Unlike NFSv2 and NFSv3, which are stateless protocols, the NFSv4 protocol maintains context or states for the client systems which access the file server and for the locks being held. The protocol allows client recovery of locking state after a server or client failure or network partition.

A client system is identified by its hostname or IP address for NFSv2 and NFSv3. NFSv4 is identified by a unique client ID and is assigned a state identification. The VNX allows you to:

- List client states
- Display the attributes of a particular client state
- Release a client state

The release of a client state breaks all file locks, recalls all delegations the client might be holding, and closes all files opened by that client.

Manage NFSv4 on page 72 describes how to manage client states by using the -v4 -client option on the server_nfs command.

Limiting access to NFSv4 clients only

Because the security measures available in NFSv2 and NFSv3 are less stringent than those available in NFSv4, the system allows you to specify the use of NFSv4 for only an exported file system. This feature prevents a client from accessing a file system with a less secure NFS version. Specify NFSv4 access on page 39 describes how to specify the NFSv4 option on the server_export command.

CHAPTER 3

Configuring

The tasks to configure NFS are:

- Getting started 26
- Create a mount point 30
- Mount a file system 31
- Export a file system 33
- Mount a NFS share on MAC OS using the GUI 41
- Mount a NFS share on MAC OS manually 42
- Configure secure NFS by using a Windows Kerberos KDC 42
- Configure secure NFS by using a UNIX or Linux Kerberos KDC 50
- Use the Solaris mapping utility gsscred 59
- Mount a VNX file system on a client system 63

Getting started

If you are using NFSv2 or NFSv3, you do not need to perform any prerequisite tasks. By default, NFSv3 is the active NFS service on the VNX. However, if you are planning to use NFSv4, perform the following setup procedures:

- Install client: Install Fedora 15 or any other intended operating system on the client system.
- Enable NFSv4 on page 26
- Configure the NFSv4 domain on page 26
- Create a user file system on page 28
- Configure access to naming services on page 28
- Verify or Enable Unicode on page 29

Enable NFSv4

To enable NFSv4 support on the VNX:

Procedure

1. With a text editor, open the /nas/server/slot_‹x›/config file, where ‹x› is the slot number of the Data Mover.
2. On the nfs config line, append the option hivers=4 or, if the option already appears, be sure the value is set to 4. Save and close the file.

   For example: 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 can be the same as the local DNS domain name or it can be unique for NFSv4. The NFSv4 domain is only used for user and group mapping.

Use the domain parameter to specify the NFSv4 domain name. The *Parameters Guide for VNX for File* provides additional information about the domain parameter.

Note

The NFSv4 server and all NFSv4 clients that access the server must use the same NFSv4 domain. Parameter and facility names are case-sensitive.

EMC VNX Command Line Interface Reference for File provides detailed information about configuring the NFSv4 domain on a VDM in a multidomain environment.

To configure the NFSv4 domain:

- Specify the NFSv4 domain name on page 27
- Verify the domain parameter on page 27
- List the NFSv4 parameters on page 27

Specify the NFSv4 domain name

Procedure

1. To specify the NFSv4 domain name, use 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 emc.com, type:

   $ server_param server_2 -facility nfsv4 -modify domain -value emc.com

   Output:

   ```
   server_2 : done
   ```

Verify the domain parameter

Procedure

1. To verify whether the domain parameter was 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           = 'emc.com'
   configured_value        = emc.com
   user_action             = none
   change_effective        = immediate
   range                   = '*'
   description             = Sets the NFS v4 domain
   ```

List the NFSv4 parameters

Procedure

1. 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     emc.com
vnodePercent     nfsv4      80       10        10
32bitClient      nfsv4      1        1
```

Create a user file system

Managing Volumes and File Systems with VNX Automatic Volume Management provides more information about file systems.

Procedure

1. To create a user file system, use this command syntax:

   ```
   $ nas_fs -name <name> -create size= <integer>[T|G|M]
   pool=<pool>
   ```

   where:

   ‹name› = name for the file system

   ‹size› = required size of the file system in terabytes (T), gigabytes (G), or megabytes (M)

   ‹pool› = name of the AVM storage pool

   Example:

   To create a user file system named ufs1, type:

   ```
   $ nas_fs -name ufs1 -create size=100M pool=symm_std -o slice=y
   ```

   Output:

   ```
   id = 261
   name = ufs1
   acl = 0
   in_use = False
   type = uxfs
   volume = v1116
   pool = symm_std
   member_of = root_avm_fs_group_1
   rw_servers=
   ro_servers=
   rw_vdms =
   ro_vdms =
   stor_devs = 000187940268-0074,000187940268-0075,000187940268-0076,
   000187940268-0077,000187940268-0078,000187940268-0079,000187940268-
   007A,000187940268-007B
   disks = d113,d114,d115,d116,d117,d118,d119,d120
   ```

Configure access to naming services

Each Data Mover on VNX needs a mechanism for looking up user and system information, including usernames, passwords, home directories, groups, hostnames, IP addresses, and netgroup definitions. The Data Mover obtains this information by making queries to naming services.

In dual stack domains with both IPv4 and IPv6 DNS servers configured, you should ensure that at least one interface for each address type is configured on all NFS servers for a Data Mover in that domain.

You can configure one or more of the following naming services for each Data Mover:

- Local files (passwd, group, hosts, and netgroup)
- Network Information Service (NIS)
- Domain Name System (DNS) for hostname and IP address resolution
- Sun Java System Directory Server LDAP

Note

In environments with users and groups with non-ASCII names, LDAP should be used.

Configuring VNX Naming Services provides more information about configuring and managing naming services.

Note

The VNX and all NFSv4 clients that access the VNX must have access to the same naming service. If you use local files, the Data Mover's passwd and group files must share the same content as that used by the client.

Failure to configure the NFSv4 domain or a naming service to resolve user and group names results in:

- Inability to create files or directories
- Denied access
- Ownership of file system objects set to nobody/nobody

For the VDM NFS multiple domain solution, the VDM is configured to work for a naming space different than the Data Mover. The domains required by one or more VDMs loaded in the Data Mover must be configured in the Data Mover client resolvers. The VDM domain configuration is enabled to confine the name resolution to the VDM's naming space.

Similar to the Data Mover, the VDM may use local files as a source of resolution for hostnames, users, groups, and netgroups. The regular hosts, passwd, group, netgroup and nsswitch.conf files can be put in the .etc directory of the VDM root file system using the server_file command. These files are taken into account only when the VDM domain configuration is enabled with the server_nsdomains command. Otherwise, the Data Mover files are used by the VDM that functions in the same naming space as the Data Mover.

Configuring Virtual Data Movers on VNX provides more information about the VDM domain configuration and Data Mover local files setup.

Verify or Enable Unicode

NFSv4 requires UTF-8 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 Unicode and UTF-8.

If you are installing a new VNX, you can enable Unicode during the installation process before creating user files and directories. If you have an existing VNX, you must enable Unicode and then convert the files and directories before accessing data through NFSv4 clients. You also need to ensure that the proper translation directories are created and the required character translation files are installed. You can enable Unicode for one or

more Data Movers by using the command $ /nas/sbin/uc_config -on -mover ‹movername›.

Using International Character Sets on VNX for File describes how to configure and manage Unicode.

⚠ CAUTION

Once Unicode is enabled, you cannot disable it and return to ASCII mode.

Procedure

1. To verify if your system has Unicode enabled, use this command syntax to list the internationalization (I18N) mode:

   ```
   $ server_cifs <movername>|head
   ```

   where:

   ‹movername› = name of the Data Mover.

   Example:

   To list the I18N mode on server_2, type:

   ```
   $ server_cifs server_2|head
   ```

   Output:

   ```
   server_2 :
   256 Cifs threads started
   Security mode = NT
   Max protocol = SMB2
   I18N mode = UNICODE
   Home Directory Shares ENABLED, map=/.etc/homedir
   Usermapper auto broadcast enabled
   Usermapper[0] = [127.0.0.1] state:active (auto discovered)
   ```

Create a mount point

To mount an exported VNX file system, you must create a mount point (an empty directory) for that file system on a Data Mover.

Note

This step is optional. The server_mount command creates a mount point.

Procedure

1. To create a mount point for a file system, use this command syntax:

   ```
   $ server_mountpoint <movername> -create <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹pathname› = path to the mount point

   Example:

   To create a mount point called /usf1 on server_2, type:

   ```
   $ server_mountpoint server_2 -create /ufs1
   ```

   Output:

   ```
   server_2 : done
   ```

List the mount point

Procedure

1. To verify whether the mount point was created, use this command syntax:

   ```
   $ server_mountpoint <movername> -list
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To verify whether the mount point was created, type:

   ```
   $ server_mountpoint server_2 -list
   ```

   Output:

   ```
   server_2:
   /.etc_common
   /ufs1
   /ufs1_ckpt1
   /ufs2
   /ufs3
   ```

Mount a file system

After you create a mount point, mount the file system onto it by using the server_mount command.

Note

When a file system is mounted, it is integrated into the local directory tree. File systems are mounted permanently by default. If you unmount a file system temporarily and then restart the file server, the file system is remounted automatically.

To mount a file system:

- Mount a file system for basic NFS access on page 31
- Mount a file system for NFSv4 access on page 32
- Unmount a file system on page 33

Mount a file system for basic NFS access

Use the server_mount command to mount a file system for basic NFS access. By default, the server_mount command mounts the file system by using the NATIVE access policy. A mount point must begin with a forward slash (/).

Procedure

1. 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 mount point for the Data Mover

Example:

To mount a file system on server_2, type:

```
$ server_mount server_2 ufs1 /ufs1
```

Output:

```
server_2 : done
```

Verify the mount point

Procedure

1. To verify whether the mount was created, use this command syntax:

   ```
   $ server_mount <movername>
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To verify whether the mount was created, type:

   ```
   $ server_mount server_2
   ```

   Output:

   ```
   server_2 :
   root_fs_2 on /    uxfs,perm,rw
   root_fs_common on / .etc_common uxfs,perm,ro
   ufs1 on /ufs1 uxfs,perm,rw
   ```

Mount a file system for NFSv4 access

Use the server_mount command to mount a file system for NFSv4 access. Planning considerations on page 18 provides more information on configuring the server_mount options. 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

Procedure

1. 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

   Examples:

   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
```

Unmount a file system

You can unmount file systems from a Data Mover temporarily or permanently by using the server_umount command. Unexporting NFS exports of the file systems (by using the server_export -unexport option) before you unmount all file systems on the Data Mover is recommended, particularly when you are unmounting file systems permanently. To change the way a file system is mounted, use the server_umount command to unmount the file system permanently on the Data Mover and then remount the file system.

Note

When you permanently unmount a file system, it is recommended that you run the server_mountpoint command to delete the old mount point.

Managing Volumes and File Systems for VNX Manually provides more information about unmounting the file systems.

Note

The server_umount command recalls all NFSv4 delegations and breaks all outstanding locks. It might take several seconds to recall file delegations.

Export a file system

To make VNX file system available to NFS users, you must export a path to the file system by using the server_export command.

Each time the server_export command is issued, an entry is added to existing entries in an export table. Entries to the table are permanent and are automatically reexported if the system restarts.

You can specify a number of options for the NFS clients of each exported file system by using the -option argument, including:

- Levels of read and write access
- Restricted access by VLAN
- Exclusive NFSv4 access
- User authentication method

Planning considerations on page 18 provides more information on understanding these options.

Exporting a file system consists of the following tasks:

- Export a file system for basic NFS access on page 34
- Levels of access on page 35
- Multiple levels of a file system on page 37
- Prevent setuid and setgid bit access on page 36
- Restrict access by VLAN on page 38
- Specify NFSv4 access on page 39

- Specify the user authentication method on page 39

Note

The server_export command described in the procedure below can be applied on a VDM as the *movername*. *Configuring Virtual Data Movers on VNX* provides more information about exporting a file system for a VDM.

Export a file system for basic NFS access

Use the server_export command to export a file system for basic NFS access. By default, the server_export command exports the file system as read-write for all NFS clients and uses UNIX user authentication.

Procedure

1. To export VNX file system for NFS access by client systems, use this command syntax:

   ```
   $ server_export <movername>-Protocol nfs [-name <name>]
   [-ignore] [-option <options>] [-comment <comment>]
   <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹name› = optional name for the NFS export

   ‹options› = options to be applied to the NFS export

   ‹comment› = optional comment

   ‹pathname› = NFS export pathname

   Examples:

   To export a file system for NFS access, type:

   ```
   $ server_export server_2 -Protocol nfs -name nasdocsfs -comment
   'NFS Export for ufs1' /ufs1
   ```

   To export all file systems for NFS access, type:

   ```
   $ server_export server_2 -Protocol nfs -all
   ```

   Output:

   ```
   server_2: done
   ```

Verify the file system exported

Procedure

1. To verify whether the file system was exported, use this command syntax:

   ```
   $ server_export <movername> -list <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹pathname› = pathname of the NFS export. If no pathname is specified, all file system exports and shares are listed.

   Example:

   To verify whether the file system was exported, type:

   ```
   $ server_export server_2 -list /ufs1
   ```

Output:

```
server_2 :
export "/ufs1" name=/nasdocsfs comment="NFS Export for ufs1"
```

Levels of access

Use the -option argument on the server_export command to specify the level of read and write access available to the NFS clients of each exported file system. Clients are identified by their hostname, netgroup, subnet, or IP address.

Specify individual user access by using either mode bits or ACLs. Planning considerations on page 18 provides more information on understanding these options.

NFS export access options on page 35 describes the access options.

Table 3 NFS export access options

Options	Description
ro	Exports the path for all NFS clients as read-only.
ro=‹clients›	Exports the path for specified NFS clients as read-only.
rw=‹clients›	Exports the path as read or write for specified clients. If no other options are specified, all clients will have read-only access.
access=‹clients›	Provides default access for the specified clients. Denies access to those NFS clients who are not given explicit access.
root=‹clients›	Provides root access to clients listed in the export command, by specifying a client for root=. Setting root access does not grant access to the export by itself. Root access is added to the other permissions.

Client lists for ro=, rw=, access=, and root= can be a hostname, netgroup, subnet, or IP address and must be colon separated, without spaces. You can also exclude access by using the dash (-) prior to an entry for ro=, rw=, and access=. For example, rw=-host1.

The maximum length of the full export string is 8184 bytes for regular releases and the maximum length for each option listed in NFS export access options on page 35 is 2048 bytes.

Note

The length of the full string is reduced to 8160 bytes for diagnostic builds.

System access behavior on page 91 provides detailed information on how to set and interpret the access options.

Note

If you are planning to define a user authentication method by using the sec option, you must specify it before defining an access option. Otherwise, the export fails. User authentication behavior on page 97 describes the rules to observe when defining the security options.

Specify levels of access

Procedure

1. To specify an access level on an exported file system, use this command syntax:

```
$ server_export movername -Protocol nfs [-option <options>]
<pathname>
```

where:

‹movername› = name of the Data Mover

‹options› = export options, separated by commas

‹pathname› = named path

Examples:

To export the file system /ufs1 as read-only for all NFS clients, type:

```
$ server_export server_2 -Protocol nfs -option ro /ufs1
```

To export the file system /ufs1 as read-only for specified NFS clients, type:

```
$ server_export server_2 -Protocol nfs -option ro,
access=172.24.102.0/255.255.255.0 /ufs1
```

To export the file system /ufs1 as read-only and providing root privileges for a specified NFS client, type:

```
$ server_export server_2 -Protocol nfs -option ro,
root=172.24.102.240, access=172.24.102.0/255.255.255.0 /ufs1
```

Output:

```
server_2 : done
```

Prevent setuid and setgid bit access

The setuid and setgid bits are UNIX permission bits that allow an executable to be run with the privileges of a different user without being authenticated as that user. The nosuid option prevents users on specified NFS clients from setting the UNIX setuid and setgid bits on any file on the exported pathname (either directly or by copying or creating a file with the setuid or getuid permissions bits set), regardless of the mount options set on the client.

When the nosuid NFS export option is used with a list of client names, the setuid and setgid bits are cleared from the permissions before setting the permissions on any file on the exported pathname for those clients. When the nosuid NFS export option is used with a dash (-) before each client name, the setuid and setgid bits are cleared from the permissions before setting the permissions on any file on the exported pathname for all clients except for the clients listed.

Procedure

1. To restrict the setuid and setgid bit access, use one of the following command syntaxes:

   ```
   $ server_export <movername> -Protocol nfs
   [-option nosuid=<client>[:<client>]...] <pathname>

   OR

   $ server_export <movername> -Protocol nfs
   [-option nosuid=<-client>[:<-client>]...] <pathname>
   ```

   Where:

   ‹movername› = name of the Data Mover

   ‹client› = name of the client or clients for which to export files, separated by colons

‹pathname› = named path

Example:

To export the NFS pathname "/users/gary" on Data Mover server_2 restricting setuid and setgid bit access for clients host10 and host11, type:

```
$ server_export server_2 -Protocol nfs -option nosuid=host10:host11
/users/gary
```

Output:

```
server_2 : done
```

Example:

To export the NFS pathname "/production1" on all Data Movers restricting setuid and setgid bit access for client host123, type:

```
$ server_export ALL -option nosuid=host123 /production1
```

Output:

```
server_2 : done
```

Example:

To export the NFS pathname "/fs1" on all Data Movers restricting setuid and setgid bit access for all clients except for 10.241.216.239, which is allowed root privileges in addition to setuid and setgid bit access, type:

```
$ server_export server_2 -Protocol nfs -option
root=10.241.216.239,nosuid=-10.241.216.239 /fs1
```

Output:

```
server_2 : done
```

Multiple levels of a file system

Use the server_export command to export multiple levels of a file system's directories including subdirectories under an already exported path. Tree quotas are useful for setting this up.

Access permissions (level of read and write access) available to the NFS clients of each exported file system might be defined differently for each export. Conflicting access permissions are handled as follows:

- For NFSv2 and NFSv3, clients that mount an exported path get the same access permissions to the entire path even if a subdirectory of that path is exported with some other permissions. Clients get the access permissions of the exported subdirectory only if they mount the subdirectory and access that mount point.
- For NFSv2 and NFSv3, clients handle conflicting access permissions in a way that deviates from the NFSv2 and NFSv3 protocol standard.
- For NFSv4, access permissions are calculated every time a client crosses a directory.

For example, if an NFSv2 and NFSv3 client is given read-only access to /ufs1, and read-write access to /ufs1/dir1, and that client mounts /ufs1, all the data in /ufs1 is read-only, including /ufs1/dir1. If the client mounts /ufs1/dir1, all the data in /ufs1/dir1 is read-write.

An NFSv4 client always accesses the pseudo-root of the server and then navigates until it locates the file system it is trying to mount. NFSv4 does not have real mounts and access permissions are verified every time a directory is crossed.

Export multiple levels of a file system

Procedure

1. To export multiple levels of VNX file system for NFS access by client systems, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs [-name <name>]
   [-ignore] [-option <options>][-comment <comment>]
   <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹name› = optional name for the NFS export

   ‹options› = options to be applied to the NFS export

   ‹comment› = optional comment

   ‹pathname› = NFS export pathname

   Examples:

   To export a file system by giving read or write access to all clients, type:

   ```
   $ server_export server_2 -Protocol nfs /ufs1
   ```

   To export a subdirectory of the previously exported path, by giving read only access to clients, type:

   ```
   $ server_export server_2 -Protocol nfs -option ro /ufs1/dir1
   ```

   Output:

   ```
   server_2: done
   ```

Restrict access by VLAN

You can limit exported file system access to hosts that belong to a specific VLAN by setting the vlan option. Hosts on other VLANS are denied access. You can specify a single VLAN or multiple VLANs.

Note

Only NFS users with UNIX security (AUTH_SYS) can use this procedure to restrict exported file system access to hosts. This procedure cannot be used by users or systems with Kerberos authentication.

Procedure

1. To export VNX file system for NFS access by client systems included in the specified VLANs, 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 by client systems included in the specified VLANs, type:

```
$ server_export server_2 -Protocol nfs -option vlan=102,103,104
```

Output:

```
server_2: done
```

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 setting the hivers option for the nfs start command in the netd file, 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.

Note

If you plan to define a user authentication method by using the sec option, you must specify it before defining the nfsv4only option. Otherwise, the export fails. User authentication behavior on page 97 describes the rules to observe when defining the security options.

Procedure

1. To export VNX file system for NFS 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 server_2 -Protocol nfs -option nfsv4only /ufs1
   ```

   Output:

   ```
   server_2: done
   ```

Specify the user authentication method

You specify the method by which the NFS service authenticates users when you export a file system.

The user authentication method is one of the following:

- UNIX security
- Kerberos

UNIX security, also referred to as AUTH_SYS security, is the default authentication method used by all standard NFS clients and servers. It is specified by using the security option sys.

Secure NFS provides Kerberos-based user and data authorization, integrity, and privacy. It supports and specifies the following security options:

- krb5 — Kerberos user and data authentication.
- krb5i — Kerberos authentication and data integrity by adding a signature to each NFS packet.
- krb5p — Kerberos authentication and data privacy by encrypting the data before sending it over the network. Data encryption requires additional resources for system processing.

User authentication on page 19 provides more information on these options.

Note

First configure NFS by using AUTH_SYS. After the NFS service is running successfully, configure secure NFS.

Secure NFS must be established before you can export a file system by using Kerberos authentication. Configure secure NFS using a Windows Kerberos KDC on page 42 and Configure secure NFS using a UNIX or Linux Kerberos KDC on page 50 provide more information on configuring secure NFS.

Note

All the NFS export options for VDM are supported.

Export a file system for Kerberos authentication

By default, the server_export command exports the file system as read-write for all NFS clients. No access is allowed for users who present UNIX credentials.

Procedure

1. To export VNX file system with Kerberos authentication, 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 with Kerberos authentication, type:

   ```
   $ server_export lngbe245 -Protocol nfs -option sec=krb5 /ufs1
   ```

   Output:

   ```
   lngbe245: done
   ```

Export a file system for UNIX and Kerberos authentication

In this example, users who authenticate with UNIX credentials have read-only access. Users who authenticate through Kerberos have read or write access to the file system.

Procedure

1. To export VNX file system with UNIX and Kerberos authentication, 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 by using UNIX and Kerberos authentication, type:

```
$ server_export lngbe245 -Protocol nfs -option
sec=krb5,sec=sys:ro /ufs1
```

Output:

```
lngbe245: done
```

Mount a NFS share on MAC OS using the GUI

Before you begin

- Click **Finder** and select **Go** › **Utilities** › **Terminal**
- Login with the admin password.

Join MAC to the Windows domain

1. Install the Administration tools disk (applicable to MAC OS Client ONLY).
2. Open the Network configuration. Click the **Apple Icon** › **System Preferences** › **Network**.
3. Click **Advanced** button and select the **DNS** tab.
4. Set Windows DNS IP addresses and set **Search domain** to your Windows domain.
5. From **Application Utilities** Windows, start the **Directory Utility**.
6. Click **Finder** and select **Go** › **Utilities** › **Terminal**.
7. To make any changes, unlock the Directory Services by clicking on the lock icon.

   Note

   These changes must be made by the local MAC client administrator.
8. In **Services**, double-click on **Active Directory** to join a domain.
9. Fill the Active Directory information and click **OK**.
10. Verify that the **Search policy** includes the Active Directory with **custom path** option in the search field. Move “Active Directory” in first not grey position.

Mount a NFS share

1. Click **Go** › **Connect to Server**.
2. In the **Server Address** field, enter nfs:// *‹servername›* / *‹sharename›* and click **Connect**.
3. A login screen appears. Enter the username and password and click **Connect**.
4. You can now browse the NFS share on the Desktop.

Note

The default mountpoint is /Volumes.

Mount a NFS share on MAC OS manually

Before you begin

- Click **Finder** and select **Go** › **Utilities** › **Terminal**.
- Login with the admin password.

Note

Contrary to the CIFS share, a NFS share can be mounted on several mountpoints. The default mountpoint is /Volumes.

Procedure

1. To mount a NFS share, use this command syntax:

   `/sbin/mount-t nfs <servername>:/<sharename>/<mountpoint>`

   where:

   ‹servername› = name of the server

   ‹sharename› = name of the NFS share

   ‹mount_point› = path to mount point for the server

   Example:

   To mount a NFS share on nsx0304el1, type:

   `/sbin/mount -t nfs nsx0304el1:/server4fs2/4fs2`

   Note

   You may receive the following errors:

   mount_nfs : cannot access ‹sharename›: No route to host - check that ‹sharename› is correct.

   mount : realpath /mountpoint: No such file or directory - the mountpoint path is incorrect.

   Output:

   ```
   nsx0304el1 : done
   ```

Configure secure NFS by using a Windows Kerberos KDC

Before you begin

Secure NFS is supported by using one of the following:

- Windows Kerberos KDC
- UNIX Kerberos KDC
- Linux Kerberos KDC

Planning considerations on page 18 provides more information on these options.

Before configuring secure NFS in a Windows environment, you must configure access to a Windows 2000 domain or later, as well as to a time server and a naming service, preferably NIS:

- The Data Movers, NFS clients, and KDCs must all have system times nearly the same, that is, all the clocks in the realm must be synchronized to enable Kerberos

authentication to occur. *Configuring Time Services on VNX* describes how to configure time services.

- For the resolution of users to UIDs and groups to GIDs, EMC recommends using NIS or iPlanet with secure NFS. The Kerberos KDC, all clients, and all Data Movers must be members of the same NIS or iPlanet domain. If neither NIS nor iPlanet is used, you must add all usernames to the /etc/passwd file and groups to the /etc/group file on the Data Mover. *Configuring VNX Naming Services* describes how to configure access to NIS and iPlanet.

To configure secure NFS by using a Windows KDC:

- Enable dynamic updates of reverse lookups on page 43
- Configure CIFS on page 43
- Set the secure NFS service instance on page 46
- Set the NFS user and group mapping service on page 48
- Reorder ACEs (multiprotocol environment only) on page 49

Enable dynamic updates of reverse lookups

Set the updatePTRrecord parameter to instruct the Data Mover's DNS client to dynamically update the PTR record for all CIFS servers.

DNS maintains mappings of hostnames to IP addresses. It can also provide reverse lookups by maintaining pointer (PTR) records that map an IP address to a hostname.

The *Parameters Guide for VNX for File* provides additional information about the updatePTRrecord parameter.

PTR records can also be updated manually instead of using the updatePTRrecord parameter. Parameter and facility names are case-sensitive.

Procedure

1. To instruct the Data Mover's DNS client to update the PTR record for all CIFS servers, use this command syntax:

   ```
   $ server_param <movername> -facility dns -modify
   updatePTRrecord -value 1
   ```

   where:

   <movername>= name of the Data Mover

   Example:

   To instruct the Data Mover's DNS client to update the PTR record for all CIFS servers, type:

   ```
   $ server_param server_2 -facility dns -modify updatePTRrecord
   -value 1
   ```

   Output:

   ```
   server_2 : done
   ```

Configure CIFS

When using a Windows KDC, you must configure CIFS on the Data Mover to access the KDC:

Procedure

1. Create a CIFS server on the Data Mover by using this command syntax:

```
$ server_cifs <movername> -add
compname=<comp_name>,domain=<full_domain_name>
```

where:

<movername>= name of the Data Mover

<comp_name>= name of the server to be registered in DNS

<full_domain_name>= full domain name to which the server belongs. This means the name must contain at least one dot (.).

Example:

To create CIFS server cifsserver on server_2, type:

```
$ server_cifs server_2 -add compname=cifsserver,domain=emc.com
```

Note

The *EMC VNX Command Line Interface Reference for File* describes the additional server_cifs command options. *Configuring and Managing CIFS on VNX* provides more information on CIFS configuration.

2. Start the CIFS service to activate the CIFS protocol for the Data Mover by using this command syntax:

```
$ server_setup <movername> -Protocol cifs -option
start[=<n>]
```

where:

<movername>= name of the Data Mover

<n>= number of threads for CIFS users. (If there is 1 GB of memory on the Data Mover, the default is 96 threads. However, if there is more than 1 GB of memory, the default number of threads is 256.)

Example:

To start the CIFS service on server_2, type:

```
$ server_setup server_2 -Protocol cifs -option start
```

Note

The secure NFS service does not require additional CIFS threads.

3. If not joined yet, join the CIFS server to a Windows domain by using this command syntax:

```
$ server_cifs <movername> -Join
compname=<comp_name>,domain=<full_domain_name>,admin=<domain
_administrator_name>,ou=<organizational_unit>
```

where:

<movername>= name of the Data Mover

<comp_name>= name of the server to be registered in DNS

<full_domain_name>= full domain name to which the server belongs. This means the name must contain at least one dot (.).

<domain_administrator_name>= login name of the user with the right to create and manage computer accounts in the organizational unit that the CIFS server is being joined to.

<organizational_unit>= Specifies the organizational unit or container where computer accounts are created in the Active Directory. By default, computer accounts are created in an organizational unit called Computers.

Example:

To join cifsserver into the Active Directory domain emc.com by using the administrator account, and to add this server to Engineering or Computers organizational unit, type:

```
$ server_cifs server_2 -Join
compname=cifsserver,domain=emc.com,admin=administrator,ou="ou=Compu
ters:ou=Engineering"
```

4. Add the NFS service to the computer account by using this command syntax:

   ```
   $ server_cifs <movername> -Join
   compname=<comp_name>,domain=<full_domain_name>,admin=<domain
   _administrator_name>,ou=<organizational_unit> -option
   addservice=nfs
   ```

   where:

   ‹movername›= name of the Data Mover

   ‹comp_name›= name of the server to be registered in DNS

   ‹full_domain_name›= full domain name to which the server belongs. This means the name must contain at least one dot (.).

   ‹domain_administrator_name›= login name of the user with the right to create and manage computer accounts in the organizational unit that the CIFS server is being joined to.

   ‹organizational_unit›= Specifies the organizational unit or container where computer accounts are created in the Active Directory. By default, computer accounts are created in an organizational unit called Computers.

   Example:

   To add the NFS service to the server previously joined to the Windows domain, type:

   ```
   $ server_cifs server_2 -Join compname=cifsserver,domain=
   emc.com,admin=administrator -option addservice=nfs
   ```

5. Create a mount point by using this command syntax:

   ```
   $ server_mountpoint <movername> -create <pathname>
   ```

   where:

   ‹movername›= name of the Data Mover

   ‹pathname›= path to the mount point

   Example:

   To create a mount point called /usf1 on server_2, type:

   ```
   $ server_mountpoint server_2 -create /ufs1
   ```

   Note

   This step is optional. The server_mount command creates a mount point.

6. After you create a mount point, mount the file system by using 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 mount point for the Data Mover

Example:

To mount the file system ufs1 on server_2, type:

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

7. To make the file system available to NFS users, export a path to the file system. Create a file system for NFS access by using this command syntax:

   ```
   $ server_export <movername> -Protocol nfs <pathname>
   ```
   where:

   <movername> = name of the Data Mover

   <pathname> = path to the mount point

   Example:

   To create a file system for NFS access, type:

   ```
   $ server_export server_2 -Protocol nfs -name /ufs1
   ```

8. (Optional) If you have an environment where you want NFS and CIFS users to access the same file system, you must create a CIFS share. Create a share for CIFS access by exporting the share's pathname by using this command syntax:

   ```
   $ server_export <movername> -Protocol cifs -name <sharename>
   <pathname>
   ```
   where:

   <movername> = name of the Data Mover

   <sharename> = name of the CIFS share

   <pathname> = path to the mount point

   Example:

   To create a share named cifs_share on server_2, type:

   ```
   $ server_export server_2 -Protocol cifs -name cifs_share /ufs1
   ```

Set the secure NFS service instance

For the secure NFS service to authenticate to Kerberos and then authenticate users, it needs a secure NFS service instance.

The secure NFS service instance is in the form of service@server, where service is NFS and server is the Data Mover hostname, for example, nfs@server_2.

When VNX is installed, a secure NFS instance is added, by default, to the secure NFS configuration file. You can use this instance or create a new instance.

View the initial secure NFS configuration

Procedure

1. To view the configuration of the secure NFS service, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs
   ```
   where:

   <movername> = name of the specified Data Mover

   Example:

   To view the configuration of the secure NFS service, type:

   ```
   $ server_nfs server_2 -secnfs
   ```

Output:

```
server_2 :
  RPCSEC_GSS server stats
  Credential count: 1
  principal: nfs@server_2
  Total number of user contexts: 0
  Current context handle: 1
```

Create a secure NFS service instance

Procedure

1. Delete the default secure NFS instance by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -principal -delete
   nfs@<server>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹server› = type of the realm

   Example:

   To delete the default secure NFS instance for server_2, type:

   ```
   $ server_nfs server_2 -secnfs -principal -delete nfs@server_2
   ```

   Output:

   ```
   server_2 : done
   ```

2. Add the new secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -principal -create
   nfs@<server>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹server› = type of the realm

   Note

   You must add the service twice by using both formats: nfs@‹Data_Mover_host_name› and nfs@‹Data_Mover_fqdn›

   Example:

   To add a secure NFS service for the new hostname, type:

   ```
   $ server_nfs server_2 -secnfs -principal -create nfs@cifsserver
   ```

   Output:

   ```
   server_2 : done
   ```

   To add a second service instance by using the server's fully qualified domain name (FQDN), type:

   ```
   $ server_nfs server_2 -secnfs -principal -create
   nfs@cifsserver.emc.com
   ```

3. Stop the secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -service -stop
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

To stop the secure NFS service for server_2, type:

```
$ server_nfs server_2 -secnfs -service -stop
```

Output:

```
server_2 : done
```

4. Start the secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -service -start
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To start the secure NFS service for server_2, type:

   ```
   $ server_nfs server_2 -secnfs -service -start
   ```

   Output:

   ```
   server_2 : done
   ```

Set the NFS user and group mapping service

This section describes the setting of the NFS user and group mapping service.

In a secure NFS environment, user authentication is based on Kerberos principal names. However, access to files and directories within the exported file system is based on UIDs and GIDs. The user's Kerberos principal name is mapped to the UID by using a mapping service. During user authentication, this mapping information is used to determine data access to Kerberos-enabled services.

When using a Windows Kerberos KDC, mapping is done automatically (the default method). When using automatic mapping, the user's UNIX name is derived from the user's Kerberos principal name. This implies that the UNIX user namespace must be the same as the Kerberos user namespace. All users are known to Kerberos and UNIX by the same name. During the user authentication process, the user's UID and primary and secondary GIDs are derived through the secure NFS mapping service. Mapping file is not created because mappings are generated automatically as needed.

Note

This mapping service should not be confused with VNX Usermapper feature that is used by CIFS.

When an NFS user authenticates through Kerberos, the realm component of the user principal is removed, and the system locates the username on the UNIX user database—NIS or local /etc/passwd file.

For example, when the principal john@myrealm.com tries to access a file system exported by using the Kerberos security option, the system authenticates the principal. Using only the username, john, the system finds the user in the password database, and retrieves the UID and primary GID of the user. Group memberships are retrieved from the group database.

Note

Configuring VNX Naming Services describes how to create password files.

Verify the mapping method

Note

The Config_Gsscred message is only seen during the initialization of secure NFS mapping.

Procedure

1. To verify the type of mapping service used by secure NFS, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -info
   ```

   where:

   <movername> = name of the Data Mover

   Example:

   To list the mapping service for server_2, type:

   ```
   $ server_nfs server_2 -secnfs -mapper -info
   ```

   Output:

   ```
   server_2:
   Current NFS user mapping configuration is:
   Config_Gsscred::initialize: Reading NFS user mapper configuration:
   /.etc/gsscred.conf.
   ..
      gsscred db = automap
      gsscred db version = None
      passwd db = NIS
   ```

Set automatic mapping

Note

A message is sent to the server_log if automatic mapping is not working.

Procedure

1. To establish automatic mapping as the mapping method, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -set -source auto
   ```

   where:

   <movername> = name of the Data Mover

   Example:

   To establish automatic mapping on server_2, type:

   ```
   $ server_nfs server_2 -secnfs -mapper -set -source auto
   ```

   Output:

   ```
   server_2: done
   ```

Reorder ACEs (multiprotocol environment only)

There is no standard way to order the ACEs that make up an ACL. UNIX clients, such as Solaris, order ACEs differently from Windows clients. In a multiprotocol environment in which file systems are accessed by both UNIX and Windows clients, the ACE order may vary depending on the protocol (NFS or CIFS) set or the modified ACL. Some applications may set ACEs in an application-specific order. Consequently, when Windows Explorer opens the ACL of a file system object and finds that the ACE order is not the one it prefers, it displays a pop-up on the Windows client to warn that the order is incorrect and

then reorders the ACEs. To instruct the Data Mover to always sort an object's ACEs in the order expected by Windows Explorer, set the acl.sortAces parameter.

The *Parameters Guide for VNX for File* provides additional information about the acl.sortAces parameter.

Sort ACEs

Note

Parameter and facility names are case-sensitive.

Procedure

1. To sort ACEs in the order expected by Windows Explorer, use this command syntax:

   ```
   $ server_param <movername> -facility cifs -modify
   acl.sortAces -value 1
   ```

   where:

   <movername> = name of the Data Mover

   Example:

   To set the cifs acl.sortAces parameter to 1, type:

   ```
   $ server_param server_2 -facility cifs -modify acl.sortAces -value
   1
   ```

   Output:

   ```
   server_2 : done
   ```

Configure secure NFS by using a UNIX or Linux Kerberos KDC

Before you begin

Secure NFS is supported by using either Windows or UNIX or Linux Kerberos KDC. Planning Considerations on page 18 provides more information on these options.

Before configuring secure NFS, you must install and configure a Kerberos KDC as well as configure access to DNS, a time server, and a naming service, preferably NIS. Consider the following:

- The Kerberos KDC can be either a Solaris Kerberos KDC that uses the Sun Enterprise Authentication Mechanism (SEAM) software or a Linux Kerberos KDC. SEAM documentation available on the Sun website or standard Linux documentation gives more information about KDC.

  Note

  KDCs from other UNIX systems have not been tested.

- After the KDC is configured, add the Host (A) resource record for the server that is hosting the Kerberos KDC to the DNS configuration. In addition, the Data Movers and NFS clients must be configured to access DNS.
- The Data Movers, NFS clients, and KDCs must have system times nearly the same, that is, all the clocks in the realm must be synchronized to enable Kerberos authentication to occur. *Configuring Time Services on VNX* describes how to configure time services.
- For the resolution of users to UIDs and groups to GIDs, EMC recommends the use of NIS or iPlanet with secure NFS. The Kerberos KDC, all clients, and all Data Movers must be members of the same NIS or LDAP domain. If neither NIS nor iPlanet is used,

you must add all usernames to the /passwd file and groups to the /group file on the Data Mover. *Configuring VNX Naming Services* describes how to configure access to NIS and iPlanet.

To configure secure NFS:

- Set the data mover name on page 51
- Configure the Kerberos realm on page 51
- Set the secure NFS service instance on page 53
- Create the NFS Kerberos service principals on page 55
- Map user principal names to UIDs on page 57

Set the Data Mover name

Each Data Mover requires a unique name in a Kerberos realm. Therefore, you might want to set the name of the Data Mover.

If you are using a Data Mover name as a hostname and there are multiple VNX systems in the same Kerberos realm, you need to change the Data Mover name.

Note

If you have a unique hostname and IP address associated with the Data Mover's interface, you do not need to set the Data Mover name. If you change a Data Mover name, ensure that you add it to the DNS server.

Procedure

1. To set the Data Mover's name, use this command syntax:

   ```
   $ server_name <movername> <new_name>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹new_name› = new name for the Data Mover

   Example:

   To rename the Data Mover server_2 to lngbe245, type:

   ```
   $ server_name server_2 lngbe245
   ```

   Output:

   ```
   server_2 : lngbe245
   ```

Configure the Kerberos realm

Add the information about the Kerberos realm and DNS domain to the Kerberos configuration on the Data Mover. This configuration information is stored in the /krb5.conf file on the Data Mover. To avoid errors, do not edit this information directly. Use the `server_kerberos` command.

Before configuring the Kerberos realm, you must:

- Specify at least one KDC and DNS domain.
- Declare the realm used for secure NFS as a default realm.

Procedure

1. To type the domain and realm information, use this command syntax:

```
$ server_kerberos <movername> -add
realm=<realm_name>,kdc=<fqdn_kdc_name>,kadmin=<kadmin_
server>,domain=<domain_name>,defaultrealm
```

where:

‹movername›= name of the Data Mover

‹realm_name›= FQDN of the Kerberos realm to be added to the KDC configuration

‹fqdn_kdc_name›= FQDN of the KDC for the realm

‹kadmin_server›= FQDN of the kadmin server

‹domain_name›= full name of the DNS domain for the realm (can omit if the realm and DNS domain are the same)

Example:

To add the domain and realm information for lngbe245, type:

```
$ server_kerberos lngbe245 -add realm=example.com,kdc=
kdc_1.lss.exam.com,kadmin=kdc_1.lss.exam.com,domain=lss.exam.com,de
faultrealm
```

Output:

```
lngbe245 : done
```

Verify the results

Procedure

1. To verify the results, use this command syntax:

   ```
   $ server_kerberos <movername> -list
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To list the domain and realm information for server_2, type:

   ```
   $ server_kerberos lngbe24 -list
   ```

   Output:

   ```
   Kerberos common attributes section:
    Supported TGS encryption types:  rc4-hmac-md5 des-cbc-md5
    Supported TKT encryption types:  rc4-hmac-md5 des-cbc-md5
    Use DNS locator:                 yes
    default_realm:                   example.com

   End of Kerberos common attributes.

   Kerberos realm configuration:

   realm name:       example.com (default realm)
    kdc:              kdc_1.lss.exam.com
    kadmin:           kdc_1.lss.exam.com
    default domain:   lss.exam.com

   End of Kerberos realm configuration.

   Kerberos domain_realm section:
    DNS domain = Kerberos realm
    .lss.exam.com = example.com
   ```

Set the secure NFS service instance

For the secure NFS service to authenticate to Kerberos and then authenticate users, it needs a secure NFS service instance.

The secure NFS service instance is in the form of service@server, where service is NFS and server is the Data Mover hostname, for example, nfs@server_2.

Note

When the VNX is installed, a secure NFS instance is added, by default, to the secure NFS configuration file. Therefore, if you change the name of the Data Mover from server_n, you must change the configuration information.

View the initial secure NFS configuration

Procedure

1. To view the configuration of the secure NFS service, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs
   ```

   where:

   ‹movername› = name of the specified Data Mover

   Example:

   To view the configuration of the secure NFS service, type:

   ```
   $ server_nfs server_2 -secnfs
   ```

   Output:

   ```
   server_2 :
     RPCSEC_GSS server stats
     Credential count: 1
     principal: nfs@server_2
     Total number of user contexts: 0
     Current context handle: 1
   ```

Change the secure NFS service instance

Use this procedure to delete and add a secure NFS instance when a Data Mover's name is changed from server_n. In this example, the Data Mover's name changes from server_2 to lngbe245:

Procedure

1. Delete the default secure NFS instance by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -principal -delete
   <service@server>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹service@server› = the type of service and the realm

   Example:

   To delete the secure NFS instance for server_2, type:

   ```
   $ server_nfs lngbe245 -secnfs -principal -delete nfs@server_2
   ```

   Output:

   ```
   lngbe245 : done
   ```

2. Add the new secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -principal -create
   nfs@<server>
   ```
   where:

   ‹movername› = name of the Data Mover

   ‹server› = type of the realm

   Note

   You must add the service twice by using both formats: nfs@‹Data_Mover_host_name› and nfs@‹Data_Mover_fqdn›

   Example:

   To add a secure NFS service for the new hostname, type:

   ```
   $ server_nfs lngbe245 -secnfs -principal -create nfs@lngbe245
   ```
   To add a second service instance that uses the server's fully qualified domain name, type:

   ```
   $ server_nfs lngbe245 -secnfs -principal -create
   lngbe245.lss.exam.com
   ```

   Output:

   ```
   lngbe245 : done
   ```

3. Stop the secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername> -secnfs -service -stop
   ```
   where:

   ‹movername› = name of the Data Mover

   Example:

   To stop the secure NFS service for lngbe245, type:

   ```
   $ server_nfs lngbe245 -secnfs -service -stop
   ```
   Output:

   ```
   lngbe245 : done
   ```

4. Start the secure NFS service by using this command syntax:

   ```
   $ server_nfs <movername>-secnfs -service -start
   ```
   where:

   ‹movername› = name of the Data Mover

   Example:

   To start the secure NFS service for lngbe245, type:

   ```
   $ server_nfs lngbe245 -secnfs -service -start
   ```
   Output:

   ```
   lngbe245 : done
   ```

Multihomed Data Movers

In installations where clients access servers through the server's network interface name instead of the server hostname, you can use a multihomed Data Mover to name the server's network interfaces. If you have a multihomed Data Mover, you need to add a secure NFS instance for each network interface.

For example, if you have a Data Mover with a hostname lngbe245 and two network interfaces known as lngbe245-1 and lngbe245-2, three secure NFS instances have to be added: nfs@lngbe245, nfs@lngbe245-1, and nfs@lngbe245-2.

Note

In this situation, an interface name means a DNS hostname. This interface name can be different from the interface name that appears when using the server_ifconfig command.

Create the NFS Kerberos service principals

Service principals are the Kerberos representation of the secure NFS service instances.

For secure NFS to work:

- Create two NFS service principals for each service instance and add them to the Kerberos principal's database. The service principals use this format:

  nfs/‹Data_Mover_host_name›

  nfs/‹Data_Mover_fqdn›
- Generate encryption or decryption security keys for these service principals and add them to the Data Mover keytab.

Add service principals to KDC

Note

This procedure is relevant only when you are using a UNIX or Linux Kerberos KDC.

Service principals are the Kerberos representation of the secure NFS service instances. The way you create these principals depends on how the Control Station accesses the KDC. To add service principals when the Control Station does not have network access to the Kerberos KDC:

- Determine if the Data Mover has a keytab file.
- Create two NFS service principals.
- Generate the associated security keys.
- Copy the keytab file to the Data Mover.

Keytab file

Determine if the Data Mover has a keytab file, and if it does, copy it to the Kerberos KDC.

Procedure

1. If the Data Mover has a keytab file (from Windows configuration), copy it (/.etc/krb5.keytab) from the Data Mover to the Kerberos KDC.

   If there is no keytab file, add the service principals as described in Create the NFS Kerberos service principals on page 55.

   Use the server_file command and FTP to copy the file (binary transfer) to the /etc directory on the Kerberos KDC.

   Note

   The *EMC VNX Command Line Interface Reference for File* provides more information about using the server_file command. *Using FTP, TFTP, and SFTP on VNX* provides information on using FTP.

Service principals

Procedure

1. To create the NFS service principals from the Kerberos KDC, use this command syntax:

   ```
   kadmin: addprinc-randkey <service_principal>
   ```

   where:

   ‹service_principal› = name of the service principal

   Example:

   To create one of the service principals, use the Data Mover name (nfs/‹Data_Mover_host_name›), and type:

   ```
   kadmin: addprinc -randkey nfs/lngbe245
   ```

   To create the other service principal, the FQDN for the Data Mover, nfs/‹Data_Mover_fqdn›, type:

   ```
   kadmin: addprinc -randkey nfs/lngbe245.lss.exam.com
   ```

   Output:

   ```
   Principal "nfs/lngbe245.lss.exam.com@example.com" created.
   ```

Verify the principals

Note

The realm is automatically added to the service principals.

Procedure

1. To verify whether the principals were added, type:

   ```
   kadmin: listprincs
   ```

   Output:

   ```
   kadmin/admin@example.com
     root/admin@example.com
     kadmin/history@example.com
     kadmin/kdc_1.lss.exam.com@example.com
     root/kdc_1.lss.exam.com@example.com changepw/
     mzolla@example.com
     admin/changepw@example.com
     kadmin/changepw@example.com
     nfs/lngbe245@example.com
     nfs/lngbe245.lss.exam.com@example.com
   ```

Generate security keys

Before generating security keys, verify that the key does not exist for the service principal. For example, to generate a key for lngbe245@example.com, if you see this entry in the keytab file, principal: nfs/lngbe245@example.com, first delete the old entry.

Note

The security keys for the service principals are added into the key table.

Procedure

1. To generate keys for each service principal and add them to the keytab file, use this command syntax:

   ```
   kadmin: ktadd -k <keytab_file_path> nfs/ <name>
   ```

   where:

‹keytab_file_path› = the location of the keytab file

‹name› = the name of the previously created service principal

Example:

To generate keys for lngbe245 and add them to tmp/krb5.keytab, type:

```
kadmin: ktadd -k /tmp/krb5.keytab nfs/lngbe245
```

Output:

```
Entry for principal nfs/lngbe245..lss.exam.com with kvno 5,
encryption type DES-CBC-CRC added to keytab WRFILE:/etc/krb5/
krb5.keytab.
```

Copy file

Copy the krb5.keytab file from Kerberos KDC to the Data Mover by using FTP (binary transfer) and the server_file command. The *EMC VNX Command Line Interface Reference for File* provides information about using the server_file command.

View the keytab file

After you create the service principals and generate the security keys, you can view the keytab information.

Procedure

1. To view the keytab information, type:

   ```
   $ server_kerberos lngbe245 -keytab
   ```

   Output:

   ```
   lngbe245 :
         Dumping keytab file

         keytab file major version = 5, minor version 2

         principal : nfs/lngbe245@example.com
         realm :example.com
         encryption type : des-cbc-crc
         principal type 1, key version : 2
              key length : 8, key : e3a4570dbfb94ce5

         principal : nfs/lngbe245.lss.exam.com@example.com
         realm :example.com
         encryption type : des-cbc-crc
         principal type 1, key version : 2
              key length : 8, key : c497d3df255ef183

         End of keytab entries.
   ```

Map user principal names to UIDs

In a secure NFS environment, user authentication is based on Kerberos principal names. However, access to files and directories within an exported VNX file system is based on UIDs and GIDs. The user's Kerberos principal name is mapped to the UID by using a mapping or naming service. During user authentication, this mapping information is used to determine data access to Kerberos-enabled services.

Mapping methods on page 58 lists the three methods available to map the principal name to the UID.

Table 4 Mapping methods

Mapping method	Use
Automatic	When all users are in the same Kerberos realm.
Mapping utility (gsscred)	In a Solaris environment and if there are multiple realms.
Local mapping file stored on the Data Mover	Only if there are no other secure NFS servers.

Automatically map user principal names to UIDs

Use automatic mapping when all users are in the same Kerberos realm. Automatic mapping is always used when using a Windows Kerberos KDC.

Automatic mapping
When an NFS user authenticates through Kerberos, the realm component of the user principal is removed, and the system locates the username on the UNIX user database—NIS or local/etc/passwd file. For example, when the principal john@myrealm.com tries to access a file system exported by using the Kerberos security option, the system authenticates the principal, and then by using only the username, john, finds the user in the password database, and retrieves the UID and primary GID of the user. Group memberships are retrieved from the group database.

Note

Configuring VNX Naming Services describes how to create password files.

When using automatic mapping, the user's UNIX name is derived from the user's Kerberos principal name. This implies that the UNIX user namespace must be the same as the Kerberos user namespace and all users are known to Kerberos and UNIX by the same name.

During the user authentication process, the user's UID and primary and secondary GIDs, are derived through the secure NFS mapping service. Mapping file is not created because mappings are generated automatically as needed.

Note

This mapping service should not be confused with the VNX Usermapper feature.

It is recommended to use automatic mapping to:

- Avoid the administrative overhead of user mapping creation (gsscred) and management.
- Reduce network traffic.
- Eliminate the possibility of users denied access to a file system because there is no mapping entry in the mapping database.

Verify mapping method

Note

The Config_Gsscred message is seen only during the initialization of secure NFS mapping.

Procedure

1. To verify the type of mapping service used by secure NFS, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -info
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To list the mapping service for lngbe245, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -info
   ```

   Output:

   ```
   lngbe245:
     Current NFS user mapping configuration is:
     Config_Gsscred::initialize: Reading NFS user mapper
   configuration:
     /.etc/gsscred.conf.
     ..
        gsscred db = automap
        gsscred db version = None
        passwd db = NIS
   ```

Use automatic mapping

Note

A message is sent to the server_log if automatic mapping is not working.

Procedure

1. To establish automatic mapping as the mapping method, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -set -source auto
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To establish automatic mapping on lngbe245, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -set -source auto
   ```

   Output:

   ```
   lngbe245: done
   ```

Use the Solaris mapping utility gsscred

In a Solaris environment, use the mapping utility gsscred to map the user principal names to UIDs. During user authentication, this mapping information is used to determine data access to Kerberos-enabled services.

Copy the mapping file

The mapping file (gsscred_db) that may be created on any Sun Kerberos client by running gsscred can be stored on an NIS server (recommended) or stored locally on the Data Mover.

Note

The gsscred man pages and the SEAM documentation available on the Sun website provide details about using this mapping service.

Mapping file location on page 60 describes what to do depending on where you store the mapping file.

Table 5 Mapping file location

If you store the file on the	Then
NIS server	Copy the mapping file (gsscred_db) to the NIS server. Changes might be necessary to the information in the configuration file (gsscred.conf) on the Data Mover.
Data Mover	Copy the mapping file (gsscred_db) to the Data Mover. No change is made to the configuration file (gsscred.conf).

Note

The gsscred.conf file is created on the first secure NFS access.

Copy the mapping file to an NIS server

After copying the mapping file (gsscred_db) to the NIS server, use this procedure to specify NIS as the mapping information source.

Note

The *Configuring VNX Naming Services* technical module provides additional information about NIS and the VNX.

Procedure

1. To specify the mapping configuration as NIS, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -set -source nis
   ```

   Output:

   ```
   lngbe245: done
   ```

2. To set the location of the password database, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -set -passwddb nis
   ```

   Output:

   ```
   lngbe245: done
   ```

3. To verify the mapping configuration for NIS, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -info
   ```

   Output:

   ```
   lngbe245:

      Current NFS user mapping configuration is:
      Config_Gsscred::initialize: Reading NFS user mapper
   configuration:
      /.etc/gsscred.conf.
   ```

```
..
   gsscred db = NIS
   gsscred db version = Solaris
   passwd db = NIS
```

4. From the NIS master server, build the NIS map for gsscred_db. NIS documentation has more information.

Copy the mapping file to a Data Mover

Use FTP to copy the mapping file (/etc/gss/gsscred_db) from the Solaris system to the Control Station.

Note

By default, if no directory is specified, the /.etc directory is used. In addition, if you copy this file to a different location, you must manually change the gsscred.conf file.

Using FTP, TFTP, and SFTP on VNX describes how to use FTP.

Note

After the file is copied, do not reboot the Data Mover.

Procedure

1. To copy the mapping file to the Data Mover, use this command syntax:

   ```
   $ server_file <movername> -put <src_file> <dst_file>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹src_file› = source file

   ‹dst_file› = destination file

   Example:

   To copy the mapping file gsscred_db to the Data Mover, type:

   ```
   $ server_file lngbe245 -put gsscred_db gsscred_db
   ```

   Output:

   ```
   lngbe245 : done
   ```

Create a local mapping file

Before you begin

Use the server_nfs ‹movername› -secnfs -mapper -mapping command to create the mapping relationships. The -mapping options (-list, -create, and -delete) are relevant only when using a local mapping file.

Use this method to create the mapping information and store it on the Data Mover if there is only one secure NFS server. However, EMC recommends using automatic mapping or the gsscred utility to create the mapping information.

Create a mapping entry

Procedure

1. To create a mapping entry, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -mapping -create
   {name=<user_name>|uid= <UID>}
   ```

where:

‹user_name› = the username

‹UID› = user ID

‹movername› = name of the Data Mover

Example:

To create a mapping entry for a user nfsuser1 in the local mapping file, type:

```
$ server_nfs lngbe245 -secnfs -mapper -mapping -create
name=nfsuser1
```

Output:

```
lngbe245 : done
```

Note

Because the format of the gsscred_db map differs for a Solaris and VNX, you cannot use this command to add a mapping entry to the Solaris map. If you do, the following message appears:

```
$ server_nfs server_x -secnfs -mapper -mapping -create
name=user2
server_x :
addUser: Cannot update a Solaris map (would mix formats)
Error 4020: server_x : failed to complete command
```

Specify the location of the mapping file

The example uses the default file pathname. If you use the default, do not specify a file pathname.

Procedure

1. To specify the location of the mapping file, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -set -source <file>
   path=<file_path>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹file_path› = path of the specified file

   Example:

   To specify the location of the mapping file, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -set -source
   file path=/.etc/gsscred_db
   ```

   Output:

   ```
   lngbe245 : done
   ```

Verify the mapping configuration

Procedure

1. To verify the mapping configuration, type:

   ```
   $ server_nfs lngbe245 -secnfs -mapper -info
   ```

Output:

```
lngbe245:
   Current NFS user mapping configuration is:
   Config_Gsscred::initialize: Reading NFS user mapper
configuration:
/.etc/gsscred.conf...
   gsscred db = File
   gsscred db version = Dart_V1
   passwd db = NIS
```

Mount a VNX file system on a client system

To allow NFS users to access a VNX file system, the exported VNX file system must be mounted on their client systems with the NFS tools available.

The NFS software documentation provides more information on mounting a file system on computers that run NFS.

CHAPTER 4

Managing

The tasks to manage NFS are:

- Manage NFS 66
- Manage NFSv4 72
- Manage secure NFS 77

Manage NFS

The tasks to manage basic NFS access to file systems are:

- Unexport the NFS path to a VNX file system on page 66
- Reexport all NFS paths on a VNX on page 67
- Disable NFS access to all file systems on a data mover on page 67
- NFS Automounter feature support on page 68

Unexport the NFS path to a VNX file system

Unexport the NFS path on the Data Mover to stop NFS client access to a VNX file system.

You can temporarily or permanently stop NFS access to a file system. By default, NFS unexports are temporary. The next time the file server restarts, the entry is automatically reexported. If you specify that the unexport be permanent, the entry is deleted from the export table.

Temporarily unexport an NFS path

Procedure

1. To temporarily unexport an NFS path to a VNX file system, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs -unexport
   <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹pathname› = the NFS export pathname

   Example:

   To temporarily unexport the /ufs1 NFS path, type:

   ```
   $ server_export server_2 -Protocol nfs -unexport /ufs1
   ```

   Output:

   ```
   server_2: done
   ```

Permanently unexport an NFS path

Procedure

1. To permanently unexport an NFS path to a VNX file system, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs -unexport -perm
   <pathname>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹pathname› = the NFS export pathname

   Example:

   To permanently unexport the /ufs1 NFS path, type:

   ```
   $ server_export server_2 -Protocol nfs -unexport -perm /ufs1
   ```

   Output:

```
server_2: done
```

Reexport all NFS paths on a VNX

You can reexport all NFS paths to the VNX file systems on a VNX while the file server is running. The operation reexports all NFS entries in the export table on the file server.

Reexporting a file system temporarily unexported from the CLI causes the new export options to be appended to the old options (before unexporting). The old and new options are used when the pathname is reexported. Use this feature when you want to reexport NFS paths you temporarily unexported.

Procedure

1. To reexport all NFS paths from a VNX, type:

   $ server_export ALL -Protocol nfs -all

   Output:

   ```
   server_2 : done
        server_3 : done
        server_4 : done
   ```

Disable NFS access to all file systems on a Data Mover

To stop NFS client access to all VNX file systems on a Data Mover, unexport the NFS paths to all the file systems at once.

You can stop NFS access to the VNX file systems temporarily or permanently. By default, NFS unexports are temporary. The next time the file server restarts, the file systems are automatically reexported. If you specify that the unexports be permanent, the entries are deleted from the export table.

Temporarily unexport all NFS paths

Procedure

1. To temporarily unexport all NFS paths on a Data Mover, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs -unexport -all
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To temporarily unexport all NFS paths on server_2, type:

   $ server_export server_2 -Protocol nfs -unexport -all

   Output:

   ```
   server_2: done
   ```

Permanently unexport all NFS paths

Procedure

1. To permanently unexport all NFS paths on a Data Mover, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs -unexport -perm
   -all
   ```

   where:

   ‹movername› = name of the Data Mover

Example:

To permanently unexport all NFS paths on server_2, type:

```
$ server_export server_2 -Protocol nfs -unexport -perm -all
```

Output:

```
server_2: done
```

NFS automounter feature support

If you use the NFS automounter feature on NFS clients, you can generate an automount map file on the VNX that contains entries for every file system permanently exported for NFS access from the file server. After creating and resolving conflicts in the file, you can copy it to the NFS clients or NIS server to use as input to their automounter configurations.

This section explains how to create an automount map file on the VNX, and how to view and resolve conflicts in the file. The documentation for the NFS clients or NIS server has instructions on how to configure and use their automounter features.

Note

In all automounter feature examples in this section, all unnecessary IP addresses were removed, including the internal Data Mover IP addresses (192.168.1.x, 192.168.2.x) and loopback IP address (127.0.0.1).

Create and save an automount map file

Procedure

1. To create an automount map file and print it to the screen, type:

   ```
   $ nas_automountmap -create
   ```

   To create and save an automount map file, use this command syntax:

   ```
   $ nas_automountmap -create -out <outfile>
   ```

   where:

   ‹outfile› = name of the output file

   Example:

   To create and save an automount map file named automountmap, type:

   ```
   $ nas_automountmap -create -out automountmap
   ```

   Note

   When you create and save automount map file output, there is no output to the screen.

2. To view the automount map file and confirm that it was saved, use this command syntax:

   ```
   $ more <outfile>
   ```

   where:

   ‹outfile› = name of the output file

   Example:

   To view the automount map file and confirm that it was saved as automountmap, type:

$ more automountmap

Output:

```
ufs1 -rw,intr,nosuid
127.0.0.1,10.172.128.47,192.168.2.2,192.168.1.2:/ufs1
ufs2 -rw,intr,nosuid
127.0.0.1,10.172.128.47,192.168.2.2,192.168.1.2:/ufs2
```

Update the automount map file

If you export additional file systems, you must update the automount map file to include the new file systems. Update the automount map file by specifying the existing version when updating the file. Use this command to create an updated automount map file.

Note

In this example, two entries are added, ufs3 and ufs4. The ufs3 entry conflicts with an existing file system, so it is renamed by adding the combination of the file system name and IP address to the entry.

Procedure

1. To create new automount map file entries, merge them with an existing automount map file ‹infile›, and save the new merged automount map file as ‹outfile›, use this command syntax:

   ```
   $ nas_automountmap -create -in <infile> -out <outfile>
   ```

   where:

   ‹infile› = name of the automount map file

   ‹outfile› = name of the new output file

   For example:

   To create new automount map file entries, merge them with an existing automount map file (automountmap), and save the new merged automount map file as automountmap1, type:

   $ nas_automountmap -create -in automountmap -out automountmap1

   Output:

   ```
   Create and save merged automount map file output. No output
   to screen.
   ```

View the updated automount map file

Procedure

1. To view the updated automount map file, use this command syntax:

   ```
   $ more <outfile>
   ```

   where:

   ‹outfile› = name of the output file

   Example:

   To view the updated automount map file, type:

   $ more automountmap1

   Output:

   ```
   ufs1 -rw,intr,suid 172.24.101.195:/ufs1
   ufs2 -rw,intr,suid 172.24.101.195:/ufs2
   ```

```
ufs3_172.24.101.195 -rw,intr,suid 172.24.101.195:/ufs3
ufs3 -rw,intr,suid 172.24.101.200:/ufs3
ufs4 -rw,intr,suid 172.24.101.200:/ufs4
```

View conflicting mount points in an automount map file

Updating an automount map file might result in conflicting entries that specify the same mount point in the file. If the automount map file contains conflicting entries, you must manually correct them by using a text editor before copying the file to NFS clients or NIS server for use in their automount configurations.

Procedure

1. To view a conflicting list of mount point entries in the automount map file ‹infile› on the screen, use this command syntax:

   ```
   $ nas_automountmap -list_conflict <infile>
   ```

   where:

   ‹infile› = name of the automount map file

   Example:

   To view a conflicting list of mount point entries, type:

   ```
   $ nas_automountmap -list_conflict automountmap1
   ```

   Output:

   Conflicting list of mount point entries sent either to screen or output file:

   ```
   Conflicting lists:
         ufs1 -rw,intr,suid 172.16.21.202:/ufs1
         ufs1_172.16.21.203 -rw,intr,suid 172.16.21.203:/ufs1
   ```

Save conflicting mount points in an automount map file

Procedure

1. To save the conflicting list of mount point entries in the new automount map file ‹outfile›, use this command syntax:

   ```
   $ nas_automountmap -list_conflict <infile> -out <outfile>
   ```

   where:

   ‹infile› = name of the automount map file

   ‹outfile› = name of the output file

   Example:

   To save the conflicting list of mount point entries in the new automount map file (automountmap2), type:

   ```
   $ nas_automountmap -list_conflict automountmap1 -out automountmap2
   ```

   Output:

   ```
   Conflicting lists:
        ufs1 -rw,intr,suid 172.16.21.202:/ufs1
        ufs1_172.16.21.203 -rw,intr,suid 172.16.21.203:/ufs1
   ```

Hide NFS exports

The export table and the mount table information is fully populated by default when responding to showmount requests regardless of the client's permissions. The forceFullShowmount parameter can be used to filter out entries if the client does not have mount permissions for the file system corresponding to that entry.

The NFS export and the mount table information returned to the NFS client is by default fully populated when responding to showmount requests regardless of the permissions for the NFS clients. This behavior can be changed through the forceFullShowmount parameter where only export and mount entries to which the NFS client has access permissions are shown. No entries are returned for exports of a NFS client that has no access.

Note

The parameter forceFullShowmount takes two values: 0 or 1, with the default being 1 (show all).

Procedure

1. To hide NFS exports, use this command syntax:

   ```
   $ server_param <movername>-facility mount -modify
   forceFullShowmount -value 0
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To hide NFS exports, type:

   ```
   $ server_param server_2 -facility mount -modify forceFullShowmount
   -value 0
   ```

   Output:

   ```
   server_2 : done
   ```

Verify information for forceFullShowmount parameter

You can verify the configured value for the forceFullShowmount parameter. Default value for the forceFullShowmount parameter is 1, and to hide the NFS exports the value is set to 0.

Procedure

1. To verify whether the forceFullShowmount was set, use this command syntax:

   ```
   $ server_param <movername>-facility mount -info
   forceFullShowmount -verbose
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To verify whether the forceFullShowmount was set, type:

   ```
   $ server_param server_2 -facility mount -info forceFullShowmount
   -verbose
   ```

   Output:

   ```
   server_2 :
   name                 = forceFullShowmount
   facility_name        = mount
   default_value        = 1
   current_value        = 0
   configured_value     = 0
   user_action          = none
   change_effective     = immediate
   range                = (0,1)
   ```

```
description             = Forces response to showmount requests to
fully populate response.
```

Manage NFSv4

Tasks to manage NFSv4 access to file systems are:

- Modify state duration on page 72
- Change the delegation recall timeout on page 72
- Modify number of usable nodes on page 73
- Display NFSv4 service status on page 73
- Stop the NFSv4 service on page 74
- Restart the NFSv4 service on page 74
- Display NFSv4 clients on page 75
- Display information about NFSv4 clients on page 75
- Release NFSv4 clients on page 76
- Support 32-bit and 64-bit NFS clients on page 77

Modify state duration

The *Parameters Guide for VNX for File* provides additional information about the leaseDuration parameter and its values. Parameter and facility names are case-sensitive.

Procedure

1. To modify the time during which the server maintains client states, use this command syntax:

   ```
   $ server_param <movername> -facility nfsv4 -modify
   leaseDuration -value <new_value>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹new_value› = This value must be less than the grace period duration specified by the lockd facility gpDuration parameter.

   Example:

   To modify the time during which the server maintains client states, type:

   ```
   $ server_param server_2 -facility nfsv4 -modify leaseDuration
   -value 20
   ```

   Output:

   ```
   server_2 : done
   ```

Change the delegation recall timeout

The *Parameters Guide for VNX for File* provides additional information about the recallTimeout parameter. Parameter and facility names are case-sensitive.

Procedure

1. To modify the time the server waits before recalling delegations, use this command syntax:

```
$ server_param <movername> -facility nfsv4 -modify
recallTimeout -value <new_value>
```

where:

movername = name of the Data Mover

new_value = value for the parameter in seconds. The range value is 5–60.

Example:

To modify the time the server waits before recalling delegations, type:

```
$ server_param server_2 -facility nfsv4 -modify recallTimeout
-value 20
```

Output:

```
server_2 : done
```

Modify number of usable nodes

The percentage ranges from 10 to 80 percent (default). The *Parameters Guide for VNX for File* provides additional information about the vnodePercent parameter and its values. Parameter and facility names are case-sensitive.

Note

Restart the Data Mover for the change to take effect.

Procedure

1. To modify the maximum percentage of nodes which the NFSv4 server utilizes, use this command syntax:

   ```
   $ server_param <movername> -facility nfsv4 -modify
   vnodePercent -value <new_value>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹new_value› = value for the parameter in seconds. The range of value is 10–80.

   Example:

   To modify the maximum percentage of nodes which the NFSv4 server utilizes to 50%, type:

   ```
   $ server_param server_2 -facility nfsv4 -modify vnodePercent -value
   50
   ```

   Output:

   ```
   server_2 : done
   ```

Display NFSv4 service status

Procedure

1. To display NFSv4 service status, use this command syntax:

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

   where:

   ‹movername› = name of the Data Mover

   Example:

To display NFSv4 service status on server_2, type:

```
$ server_nfs server_2 -v4
```

Output:

```
server_2 :
     --------------nfsv4 server status ---------------
               * Service Started *
     -------- NFSv4 Clients --------
     Confirmed Clients : 2
     UnConfirmed Clients : 0
     --------------------------------
     --------- NFSv4 State --------
     Opens : 4
     Locks : 0
     Delegations : 0
     --------------------------------
```

Stop the NFSv4 service

Stop the NFSv4 service only when trying to resolve a deadlock situation. Linux-based clients do not respond smoothly when the service is stopped. It might be necessary to remount the VNX file system on the client system. For NFSv4 clients, stopping the NFSv4 service is similar to a server shutdown or a network failure. All delegations are recalled, locks are released, files are closed, and NFSv4 client states are flushed.

Procedure

1. To stop the NFSv4 service, use this command syntax:

   ```
   $ server_nfs <movername> -v4 -service -stop
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To stop the NFSv4 service on server_2, type:

   ```
   $ server_nfs server_2 -v4 -service -stop
   ```

   Output

   ```
   server_2 : done
   ```

Restart the NFSv4 service

The NFSv4 service must be enabled before you can start it. Getting started on page 26 describes how to enable NFSv4 support. This command stops the NFSv4 service only. Other versions of NFS continue to run.

Procedure

1. To restart the NFSv4 service, enable it and use this command syntax:

   ```
   $ server_nfs <movername> -v4 -service -start
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To start the NFSv4 service on server_2, type:

   ```
   $ server_nfs server_2 -v4 -service -start
   ```

   Output:

```
server_2 : done
```

Display NFSv4 clients

Procedure

1. 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 NFS 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
   ```

Note

- The first column 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 has an established state, it is no longer listed in the command output but this does not indicate a problem.

Display information about NFSv4 clients

You can identify a client system by its hostname, IP address, or index number. Obtain the index number by using the -v4 -client -list option.

Procedure

1. To display information about a client system that has an established state with the NFSv4 server, type:

   ```
   $ server_nfs <movername> -v4 -client -info { index=<index> |
   hostname=<host> | IPaddress=<addr> }
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹index› = index number assigned to the client system

‹*host*› = hostname of the client system

‹*addr*› = IP address of the client system

Examples:

To display information about a client system identified by its index number that has an established state with the NFS service on server_2, type:

```
$ server_nfs server_2 -v4 -client -info index=0xe2400000
```

```
$ server_nfs server_2 -v4 -client -info index=0xef400000
```

Output:

```
server_2 :
     win901234 : 0xe2400000
        user: usr1 : inode# 2479

server_2 :
     10.171.2.76 : 0xef400000
        user: usr2 : inode# 2478
        user: usr1 : inode# 2477
        user: -2 : inode# 2476
```

Note

The output displays a list of connected client users and the files that are open. Files are identified by inode number. To determine the name of the file associated with a particular inode number, use a command from a UNIX-based NFS client that finds ‹path-to-start-search-from› -inum ‹inode› -print. For example, find /ufs -inum 1103 -print. Note that a file may be known by more than one name.

Release NFSv4 clients

Use this command to release the client ID of a client system that has an established state with the NFSv4 server.

Releasing a client ID releases all locks and closes all files associated with that client ID. If an NFSv4 client has created more than one client ID, the release of one client ID closes the files associated to that client ID.

Note

Only release a client state when trying to resolve a deadlock. Linux-based client systems do not respond smoothly when the state is released. It may be necessary to remount the VNX file system on the client system. For NFSv4 clients, releasing the state is similar to a server shutdown or a network failure. All delegations are recalled, locks are released, and files are closed.

Procedure

1. To release the client ID of a client system that has an established state with the NFSv4 server, use this command syntax:

   ```
   $ server_nfs <movername> -v4 -client -release
   { index=<index>| hostname=<host> | IPaddress=<addr> }
   ```

   where:

   ‹*movername*› = name of the Data Mover

   ‹*index*› = index number assigned to the client system

   ‹*host*› = hostname of the client system

   ‹*addr*› = IP address of the client system

Example:

To release a client system identified by its index number that has an established state with the NFS service on server_2, type:

```
$ server_nfs server_2 -v4 -client -release index=0xe2400000
```

Output:

```
server_2 : done
```

Support 32-bit and 64-bit NFS clients

Although the NFSv4 standard mandates support for 64-bit attributes, some NFSv4 clients (for instance, Solaris 10) do not yet support 64-bit attribute values such as inodes. Consequently, the VNX NFSv4 server defaults to 32 bits.

The *Parameters Guide for VNX for File* provides additional information about the 32bitClient parameter.

Note

Parameter and facility names are case-sensitive.

Procedure

1. To enable the NFSv4 server to support 64 bits, use this command syntax:

   ```
   $ server_param <movername> -facility nfsv4 -modify
   32bitClient -value 0
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To enable the NFSv4 server to support 64 bits, type:

   ```
   $ server_param server_2 -facility nfsv4 -modify 32bitClient -value
   0
   ```

   Output:

   ```
   server_2 : done
   ```

Manage secure NFS

Tasks to manage secure NFS are:

- View keytab entries on page 78
- Display all secure NFS service instances on page 78
- Display user attributes on page 79
- Display information about a user in a local mapping file on page 79
- Delete a user from a local mapping file on page 80
- Delete service principals on page 80
- Release authentication on page 80

View keytab entries

Procedure

1. To view the keytab information, type:

   $ server_kerberos lngbe245 -keytab

   Output:

```
lngbe245 :
Dumping keytab file

keytab file major version = 5, minor version = 2

principal: nfs/lngbe245@example.com
realm: example.com
encryption type: des-cbc-crc
principal type 1, key version: 2
     key length: 8, key: e3a4570dbfb94ce5

principal: nfs/lngbe245.lss.exam.com@example.com
realm: example.com
encryption type: des-cbc-crc
principal type 1, key version: 2
     key length: 8, key: c497d3df255ef183

End of keytab entries.
```

Display all secure NFS service instances

Procedure

1. To display all secure NFS service instances, use this command syntax:

   $ server_nfs *<movername>* -secnfs -user -list

   where:

   ‹movername› = name of the Data Mover

   Example:

   To display all secure NFS service instances on lngbe245, type:

   $ server_nfs lngbe245 -secnfs -user -list

   Output:

```
lngbe245:
RPCSEC_GSS server stats

Credential count: 2
  principal: nfs@dm112-cge0.nasdocs.emc.com
  principal: nfs@dm112-cge0

  Total number of user contexts: 1
  Current context handle: 3

  PARTIAL user contexts:
  Total PARTIAL user contexts: 0

  USED user contexts:
    principal=nfsuser1@NASDOCS.EMC.COM,
    service=nfs@dm112-cge0.nasdocs.emc.com, handle=3,
       validity=35914s
  Total USED user contexts: 1

  EXPIRED user contexts:
  Total EXPIRED user contexts: 0
```

Display user attributes

Display the attributes of an authenticated user to verify the user authentication context and to verify the UID and GID to which the Kerberos principal has been mapped.

Procedure

1. To display the attributes of an authenticated user identified by its Kerberos user or service name or its user authentication context identifier, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -user -info
   {principal=<user_principal>| handle=<handle_number>}
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹user_principal› = name of the user

   ‹handle_number› = the user's ID

   Example:

   To display the attributes of an authenticated user, type:

   $ server_nfs lngbe245 -secnfs -user -info handle=38

   Output:

   ```
   lngbe245:
   principal: nfsuser1@NASDOCS.EMC.COM
   service: nfs@dm112-cge0.nasdocs.emc.com
   handle: 38
   validity: 6073s
   GSS flags: mutl conf intg redy tran
   credential: uid=1944, inuid=1944, gid=2765
   ```

Display information about a user in a local mapping file

This procedure is relevant only when using the Solaris UNIX KDC and gsscred_db or a local mapping file. Map user principal names to UIDs on page 57 provides more information about using a local mapping file.

Procedure

1. To display a user entry in a local mapping file, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -mapper -mapping -list
   {name=<user_name>| uid=<UID>}
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹user_name› = the username

   ‹UID› = user ID

   Example:

   To display information about a user nfsuser1 in the local mapping file, type:

   $ server_nfs lngbe245 -secnfs -mapper -mapping -list name=nfsuser1

   Output:

   ```
   lngbe245:
      0401000B06092A864886F7120102020000001A7365636E66
      7375736572314073746174617465732E656D632E636F6D
      1000       nfsuser1, kerberos_v5
   ```

Delete a user from a local mapping file

This procedure is relevant only when using a Solaris UNIX KDC and gsscred_db or a local mapping file. Map user principal names to UIDS on page 57 provides more information about using a local mapping file.

Procedure

1. To delete a user entry from a local mapping file, use this command syntax:

   ```
   $ server_nfs -secnfs -mapper -mapping -delete
   {name=<user_name> | uid=<UID>}
   ```

   where:

   ‹user_name› = username

   ‹UID› = user ID

   Example:

   To delete a user nfsuser1 from the local mapping file, type:

   ```
   $ server_nfs -secnfs -mapper -mapping -delete name=nfsuser1
   ```

   Output:

   ```
   lngbe245: done
   ```

Delete service principals

When you no longer want to use secure NFS, or you want to modify secure NFS, you can delete the service principal. To start it again, use the -create option.

Procedure

1. To delete a service principal, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -principal -delete
   nas@<server>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹server› = type of the realm

   Example:

   To delete a service principal, type:

   ```
   $ server_nfs lngbe245 -secnfs -principal -delete nfs@lngbe245
   ```

   Output:

   ```
   lngbe245: done
   ```

Release authentication

You might want to release the authentication to correct a UID or GID mapping problem. Obtain a list of current users and principals that use the -list option.

Procedure

1. To release the secure NFS authentication context for a user, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -user -release
   {principal=<user_principal> | handle=<handle_number>}
   ```

   where:

‹movername› = name of the Data Mover

‹user_principal› = name of the released principal

‹handle_number› = the user ID

Example:

To release authentication for the nfsuser1, type:

```
$ server_nfs lngbe245 -secnfs -user -release
principal=nfsuser1@secnfs.eng
```

Output:

```
lngbe245: done
```

CHAPTER 5

Troubleshooting

As part of an effort to continuously improve and enhance the performance and capabilities of its product lines, EMC periodically releases new versions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

If a product does not function properly or does not function as described in this document, contact your EMC Customer Support Representative.

Problem Resolution Roadmap for VNX contains additional information about using EMC Online Support and resolving problems.

Topics included in this chapter are:

- EMC E-Lab Interoperability Navigator 84
- VNX user customized documentation 84
- Troubleshooting NFSv4 84
- Troubleshooting secure NFS 86
- Error messages 88
- NFSv4 error messages 88
- EMC Training and Professional Services 89

EMC E-Lab Interoperability Navigator

The EMC E-Lab™ Interoperability Navigator is a searchable, web-based application that provides access to EMC interoperability support matrices. It is available on EMC Online Support at http://Support.EMC.com. After logging in, in the right pane under **Product and Support Tools**, click **E-Lab Navigator**.

VNX user customized documentation

EMC provides the ability to create step-by-step planning, installation, and maintenance instructions tailored to your environment. To create VNX user customized documentation, go to: https://mydocs.emc.com/VNX.

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 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 KDC.
  - Ping from the Data Mover to the client.
  - If using NIS:
    - Ping from the Data Mover to NIS.
    - 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

Procedure

1. 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 NFS statistics

Procedure

1. To display NFS statistics, use this command syntax:

   ```
   $ server_nfs <movername> -stats
   ```

   where:

   <movername> = name of the Data Mover

   Example:

   To display NFS statistics, type:

   $ server_nfs server_2 -stats

   Output:

```
server_2:

Server nfs(v2):

proc    ncalls   %totcalls   ms/call   failures

null     10        100.0       0.0       0
getattr  0         0.0         0.0       0
setattr  0         0.0         0.0       0
root     0         0.0         0.0       0
lookup   0         0.0         0.0       0
readlink 0         0.0         0.0       0
read     0         0.0         0.0       0
wrcache  0         0.0         0.0       0
write    0         0.0         0.0       0
create   0         0.0         0.0       0
remove   0         0.0         0.0       0
rename   0         0.0         0.0       0
link     0         0.0         0.0       0
symlink  0         0.0         0.0       0
mkdir    0         0.0         0.0       0
rmdir    0         0.0         0.0       0
readdir  0         0.0         0.0       0
fsstat   0         0.0         0.0       0

Server nfs (v3):

proc    ncalls   %totcalls   ms/call   failures

v3null     4        0.4        3.0       0
v3getattr  366      38.5       0.7       0
v3setattr  5        0.5        4.0       0
v3lookup   177      18.6       0.0       0
v3access   326      34.3       0.0       0
v3readlink 0        0.0        0.0       0
v3read     4        0.4        0.0       0
v3write    12       1.3        4.7       0
v3create   10       1.1        7.2       0
```

```
v3mkdir    1        0.1       16.0      0
v3symlink  0        0.0       0.0       0
v3mknod    0        0.0       0.0       0
v3remove   8        0.8       0.5       0
v3rmdir    0        0.0       0.0       0
v3rename   1        0.1       0.0       0
v3link     0        0.0       0.0       0
v3readdir  13       1.4       0.0       0
v3readdirplus 2     0.2       0.0       0
v3fsstat   6        0.6       0.0       0
v3fsinfo   15       1.6       0.0       0
v3pathconf 0        0.0       0.0       0
v3commit   0        0.0       0.0       0

nfsv4 stats

Proc         Calls     Max RWMax    Ticks  Failed  microsec/call
----         -----     --- -----    -----  ------  -------------
Null:          103       0    0        0     0           0
v4Null:         51 7106300    0 14194070     0      278315
v4Compound: 13674   13213    0  3560727     0         260
v4Close:        3      57    0      165     0          55
v4Create:      80    4252    0   283343    12        3541
v4GetAttr:     97    9337    0    47144    22         486
v4GetFh:       50       3    0      110     0           2
v4Lookup:     035    8777    0   148343     1          72
v4Open:        37    4883    0   142310     0        3846
v4Open_Conf:   33      33    0      758     0          22
v4PutFh:      266      69    0     7309     0          27
v4PutrootFh:  531     939    0     2655     0           5
v4ReadDir:     13    8461    0    22995     0        1768
v4Remove:      79   11098    0   121760     4        1541
v4Renew:    12792      43    0   157575    14          12
v4SetAttr:    126    4182    0   176486   251        1400
v4SetClntid:   35      40    0      732     0          20
v4Clntid_Conf: 35      22    0      666     0          19
v4Write:        5   13117    0    16417     0        3283
----------  ------              -------  -----     -------
Total:      30045              18883565   304         628

Max active nfs threads: 9, bad read stream 0
Total NFS procs; 13725, time 19245, ave 1
Time in readStream 870991 usec (ave 0)
Time in writeStream 63 usec (ave 80)

Server lookupcache:

nHit   nFind   nNegadd   nChecked
1736   6603    13        1736

Server rpc:

ncalls   nBadRpcData   nDuplicates   nResends   nBadAuths
256947   1             0             0          0
```

Troubleshooting secure NFS

When you encounter problems while using secure NFS, review the following:

- Realm configuration
- Secure NFS configuration
- Encryption or decryption keys
- Data Mover server log—server_log ‹movername›
- DNS and NIS configuration on the Data Mover and the client

- NFS user mapping configuration
- Time synchronization setup
- Syntax of the server_mount command, specifying the security option
- Syntax of the server_export command
- Network traces
- Ticket cache on the client—klist
- KDC log on the client, default location is /var/krb5/kdc.log (relevant only if using a UNIX/Linux KDC)

Validate the following:

- Status of services:
  - gssd daemon
  - KDC services
  - DNS service
- Check connectivity:
  - Ping from the Data Mover to the KDC.
  - Ping from the Data Mover to the client.
  - If using NIS:
    - Ping from the Data Mover to NIS.
    - Verify that the users or groups that have problems are in the NIS passwd, group, or gsscred_db files.
- NFS client:
  - Export the given share from the Data Mover without Kerberos.
  - Mount the given share from the NFS client.

Cannot access file system

If a user cannot access a file system, ensure that the user has a valid security context.

Procedure

1. To retrieve a list of users and principals, type:

   ```
   $ server_nfs lngbe245 -secnfs -user -list
   ```

   Output:

   ```
   server_lngbe245 :
        RPCSEC_GSS server stats

      Credential count: 1
      principal: nfs@lngbe245

      Total number of user contexts: 2
      Current context handle: 5

      PARTIAL user contexts:
      Total PARTIAL user contexts: 0

      USED user contexts:
      principal=root/win901234@SECNFS.ENG, service=nfs@lngbe245,
      handle=1, validity=28028s
      principal=nfsuser1@SECNFS.ENG, service=nfs@lngbe245,
      handle=4, validity=28510s
   ```

```
Total USED user contexts: 2

EXPIRED user contexts:

Total EXPIRED user contexts: 0
```

Release user authentication

If the user is listed, try to release authentication for that user, and then have the user access the file system again.

Procedure

1. To release authentication for a user, use this command syntax:

   ```
   $ server_nfs <movername> -secnfs -user -release
   principal=<user_principal>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹user_principal› = the released principal

   Example:

   To release authentication for nsfuser1, type:

   ```
   $ server_nfs lngbe245 -secnfs -user -release
   principal=nfsuser1@secnfs.eng
   ```

   Output:

   ```
   lngbe245: done
   ```

Error messages

All event, alert, and status messages provide detailed information and recommended actions to help you troubleshoot the situation.

To view message details, use any of these methods:

- Unisphere software:
  - Right-click an event, alert, or status message and select to view Event Details, Alert Details, or Status Details.
- CLI:
  - Type nas_message -info ‹MessageID›, where ‹MessageID› is the message identification number.
- *Celerra Error Messages Guide*:
  - Use this guide to locate information about messages that are in the earlier-release message format.
- EMC Online Support:
  - Use the text from the error message's brief description or the message's ID to search the Knowledgebase on EMC Online Support. After logging in to EMC Online Support, locate the applicable **Support by Product** page, and search for the error message.

NFSv4 error messages

NFS error numbers are assigned to failed operations within a compound request. A compound request contains a number of NFS operations, the results of which are

encoded in sequence in a compound reply. The results of successful operations consist of an NFS4_OK status followed by the encoded results of the operation. If an NFS operation fails, an error status is entered in the reply and the compound request is terminated.

Note

If there are differences between these descriptions and RFC 3530, the RFC description takes precedence.

EMC Training and Professional Services

EMC Customer Education courses help you learn how EMC storage products work together within your environment to maximize your entire infrastructure investment. EMC Customer Education features online and hands-on training in state-of-the-art labs conveniently located throughout the world. EMC customer training courses are developed and delivered by EMC experts. Go to EMC Online Support at http://Support.EMC.com for course and registration information.

EMC Professional Services can help you implement your system efficiently. Consultants evaluate your business, IT processes, and technology, and recommend ways that you can leverage your information for the most benefit. From business plan to implementation, you get the experience and expertise that you need without straining your IT staff or hiring and training new personnel. Contact your EMC Customer Support Representative for more information.

APPENDIX A

System Access Behavior

This appendix describes how to interpret the read and write access that you set for NFS clients of exported file systems by using the -option argument to the server_export command. Clients are identified by their hostname, netgroup, subnet, or IP address.

Topics included are:

- Behavior when specifying combinations of access modes 92
- Rules for resolving conflicts among hosts, subnets, and netgroups 92
- Enable NFS special look up order for netgroups 95
- Specify read-only as the default access mode 95

Behavior when specifying combinations of access modes

Access mode combinations on page 92 shows what happens when specifying combinations of access modes for the server_export command. An X in the table column means that the access mode was specified.

Table 6 Access mode combinations

	ro	ro=	rw=	access=	Resulting behavior
1	X	X	X	X	Do not specify ro and ro= in the same export. The ro option is ignored. Access is the same as row 9.
2	X	X	X		Do not specify ro and ro= in the same export. The ro option is ignored. Access is the same as row 10.
3	X	X		X	Do not specify ro and ro= in the same export. The ro option is ignored. Access is the same as row 11.
4	X	X			Do not specify ro and ro= in the same export. The ro option is ignored. Access is the same as row 12.
5	X		X	X	Read-only to access list hosts, read/write to read/write list hosts. Access is denied to all other hosts.
6	X		X		Read/write to the read/write list hosts. Read-only access to all other hosts.
7	X			X	Read-only access to the access list. Access is denied to all other hosts.
8	X				Read-only access to all.
9		X	X	X	Do not specify ro= and access= in the same export. Read-only to the read-only list and access list. Read/write to the read/write list. Access is denied to all other hosts.
10		X	X		Read/write to read/write list, read-only to read-only list. Access is denied to all other hosts.
11		X		X	Do not specify ro= and access= in the same export. Read-only to the read-only and access list. Access is denied to all other hosts.
12		X			Read-only to read-only list. Access is denied to all other hosts.
13			X	X	Read/write to read/write list, read-only to access list. Access is denied to all other hosts.
14			X		Read/write to read/write list. Read-only to all other hosts.
15				X	Read/write access to the access list. Access is denied to all other hosts.
16					Read/write to all hosts.

Rules for resolving conflicts among hosts, subnets, and netgroups

If hosts belong to multiple netgroups and these netgroups can be given different types of access by using the server_export command, certain rules apply. Rules for resolving conflicts on page 93 explains these rules and provides examples. In these examples, host1 is a member of all these subnets and netgroups.

Table 7 Rules for resolving conflicts

Rule	Example	Access
By default, hosts and subnets have preference over netgroups.	rw=host1, ro=netgrp1	Host1 is granted read/write access. All other hosts belonging to netgrp1 are granted read-only access.
	rw=netgrp1, ro=subnet1	Host1 is granted read-only access. All hosts belonging to subnet1 including host1 are granted read-only access. All hosts belonging to netgrp1 except host1 are granted read or write access.
You can specify the spExportLookup parameter to set preference for hosts and netgroups over subnets. Enable NFS special lookup order for netgroups on page 95 provides information about setting this parameter.	rw=netgrp1, ro=subnet1	If the parameter is set, all hosts belonging to netgrp1 including host1 are granted read/write access. All hosts belonging to subnet1 except host1 are granted read-only access.
	rw=subnet1, ro=netgrp1	All hosts belonging to subnet1 including host1 are granted read/write access. All hosts belonging to netgrp1 except host1 are granted read-only access. If parameter is set, all hosts belonging to netgrp1 including host1 are granted read-only access. All hosts belonging to subnet1 except host1 are granted read/write access.
	rw=-subnet1:netgrp1	All hosts belonging to subnet1 including host1 are granted read-only access. All hosts belonging to netgrp1 except host1 are granted read/write access. If parameter is set, all hosts belonging to netgrp1 including host1 are granted read-write access. All hosts belonging to subnet1 except host1 are granted read-only access. **Note** Negation is used in this example.
	ro=-subnet1:netgrp1	Host1 is denied access as it is in the excluded list. If parameter is set, host1 is granted read-only access as it belongs to netgrp1. **Note** Negation is used in this example.

Table 7 Rules for resolving conflicts (continued)

Rule	Example	Access
If you use read or write and read-only access, read/write access is granted by default. To specify read-only as the default access, specify the secureExportMode parameter. Specify read only as the default access mode on page 95 provides information about setting this parameter.	rw=host1, ro=host1	Host1 is granted read or write access.
	ro=subnet1, rw=subnet1	All hosts belonging to subnet1 are granted read/write access. **Note** If the parameter is set, host1 is granted read-only access.
If you use a host and a subnet, the first matching entry is used for granting access.	rw=-subnet1:host1	Host1 is denied read/write access. All other hosts belonging to subnet1, including Host1 are denied read/write access.
	rw=host1:-subnet1	Host1 is granted read/write access. All other hosts belonging to subnet1 are denied read/write access. **Note** Negation is used in this example.
If there are two hosts, the first matching entry is used for granting access.	rw=-host1:host1	Host1 is denied read/write access.
	rw=host1:-host1	Host1 is granted read/write access. **Note** Negation is used in this example.
If there are two subnets, the first matching entry is used for granting access.	rw=-subnet1:subnet2	Host1 is denied read/write access. All other hosts belonging to subnet1 are denied read/write access. All other hosts belonging to subnet2 are granted read/write access.
	rw=subnet2:-subnet1	Host1 is granted read/write access. All other hosts belonging to subnet2 are granted read/write access. All other hosts belonging to subnet1 are denied read/write access. **Note** Negation is used in this example.
If there are two netgroups, the first matching entry is used for granting access.	rw=-netgrp1:netgrp2	Host1 is denied read/write access. All other hosts belonging to netgrp1 are denied read/write access.

Table 7 Rules for resolving conflicts (continued)

Rule	Example	Access
		All other hosts belonging to netgrp2 are granted read/write access.
	rw=netgrp2:-netgrp1	Host1 is granted read/write access. All other hosts belonging to netgrp2 are granted read/write access. All other hosts belonging to netgrp1 are denied read/write access. **Note** Negation is used in this example.

Enable NFS special look up order for netgroups

You can prioritize access permission for hosts and netgroup memberships of the NFS client over access permission for the subnets of the NFS client. By default, access permissions determined by a hostname and a subnet are given higher priority followed by permissions determined by the netgroup membership.

The *Parameters Guide for VNX for File* provides additional information about the spExportLookup parameter. Parameter and facility name are case-sensitive.

Note

By default, the parameter value is 0.

Procedure

1. To turn on spExportLookup, use this command syntax:

   ```
   $ server_param <movername> -facility nfs -modify
   spExportLookup -value 1
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To set the NFS spExportLookup paramter, type:

   ```
   $ server_param server_2 -facility nfs -modify spExportLookup -value
   1
   ```

   Output:

   ```
   server_2 : done
   ```

Specify read-only as the default access mode

When clients, subnets, or netgroups are present in a server_export -option list and there is a conflict about whether to grant read/write or read-only access, read/write access is given by default.

You can set a parameter that enables read-only access when a conflict exists by changing the secureExportMode parameter. Rules for resolving conflicts among hosts, subnets, and netgroups on page 92 provides more information.

The *Parameters Guide for VNX for File* provides additional information about the secureExportMode parameter. Parameter and facility names are case-sensitive.

Procedure

1. To turn on secure export mode, use this command syntax:

   ```
   $ server_param <movername> -facility nfs -modify
   secureExportMode -value 1
   ```

   where:

   <movername> = name of the Data Mover

   Example:

   To set the NFS secureExportMode parameter to 128, type:

   ```
   $ server_param server_2 -facility nfs -modify secureExportMode
   -value 1
   ```

   Output:

   ```
   server_2 : done
   ```

APPENDIX B

User Authentication Behavior

This appendix describes how to interpret the security you set for NFS clients of exported file systems by using the sec option to the server_export command.

Topics included are:

- General rules when specifying security options 98
- Root access mode 98

General rules when specifying security options

The rules listed in Specifying security option rules on page 98 apply when specifying the security option.

Note

Access modes contain no spaces and are separated by a comma (,).

Table 8 Specifying security option rules

Rule	Example
If the security option is not specified, the file system is exported by using the AUTH_SYS method.	`server_export lngbe245 -Protocol nfs -option ro /ufs`
If you specify an access mode before the security option, the export fails.	`server_export lngbe245 -Protocol nfs -option rw=client1:client2, sec=krb5:ro=client3 /ufs1`
If you repeat the value of a security option, the export fails.	`server_export lngbe245 -Protocol nfs -option sec=krb5 :rw=client1, sec=krb5 :ro=client2 /ufs1`
Multiple security options can be used in the same command. In this example, users authenticating with UNIX credentials, have read-only access and users authenticating with Kerberos have read/write access.	`server_export lngbe245 -Protocol nfs -option sec=krb5,sec=sys :ro / ufs1`
A single security option might have multiple modes. In this example, all clients authenticate through Kerberos. client1 and client2 have read/write access. client3 and client4 have read-only access.	`server_export lngbe245 -Protocol nfs -option sec=krb5: rw=client1:client2, ro=client3:client4 /ufs1`

Root access mode

The root= access mode applies only to root users from a specified hostname, netgroup, subnet, or IP address. The security mode does not affect the access granted to the file system. However, the type of access a root user receives depends on the security mechanism used for authentication, as explained in Root access and secure NFS on page 98.

Table 9 Root access and secure NFS

Command example	Explanation
`server_export lngbe245 -Protocol nfs -option sec=krb5:root=client1,sec=sys: ro /ufs1`	Root user on client1 can access the file system. If the root user authenticates by using Kerberos security, read/write access is granted. If the user authenticates by using the default security, read-only access is granted.
`server_export lngbe245 -Protocol nfs -option sec=krb5:root=client1, sec=sys:access=client1 /ufs1`	All clients authenticated by using Kerberos are granted read/write access.

Table 9 Root access and secure NFS (continued)

Command example	Explanation
	Root user from client1 is also granted access with root privileges.
`server_export lngbe245 -Protocol nfs -option sec=krb5:access=client1, root=client2 /ufs1`	Only client1 has access to the file system. All other clients, including client2, are denied access. For client2 to have access, it needs to be on the access list.

APPENDIX C

NFS Authentication Daemon for PC Clients

This appendix describes how to set up a PC client software and discusses the Hummingbird PC NFS client issues.

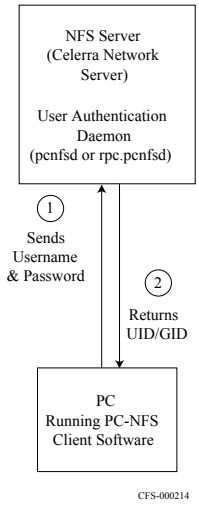
Topics included are:

- PC client access 102
- Set up PC client software 102
- Hummingbird PC NFS client issues 103

PC client access

Similar to native UNIX and Linux clients, a PC client that uses NFS to access VNX file systems must be authenticated successfully before it can get access to the VNX, as shown in PC client access on page 102.

Figure 2 PC client access



PC and UNIX environments use different NFS user authentication methods. An authentication daemon, typically rpc.pcnfsd or pcnfsd, bridges these differences.

The authentication daemon runs on a host in the customer's environment (preferred) or on the Data Mover and performs the following services:

1. The daemon receives and validates the username and password provided by the PC client.
2. The daemon assigns the PC client a user ID and group ID (UID or GID) for each user or password combination.
3. PC clients use the assigned UID or GID to access the system.

Typically, NFS software on PC clients can communicate with the authentication daemon that is running on the system without change.

If your PC software configuration does not include NFS software and you want to use the PC to access VNX file systems that uses the NFS protocol, you can purchase a PC client NFS software package, such as Hummingbird Communications Ltd.'s PC-NFS or NFS-Maestro software.

Set up PC client software

This section describes how to set up a PC client software package for network access to the VNX. The examples used are PC-NFS and NFS Maestro.

Note

- From this point on, whenever the PC client issues a mount request, the user is authenticated and the remaining activity is pure NFS traffic typical between a client and server.
- Vendor's respective user manual has details on what is required now for setup or login or both. For example, PC-NFS can detect any system that runs pcnfsd in the subnet, while NFS Maestro has an option where you can specify the name of the system.

Procedure

1. Set up a user account. server_user command has further information.

   You can also use NIS server to create a user.
2. Open the /nas/server/server_‹x›/netd file with a text editor, add pcnfs on a separate line after the nfs start line, and save the file.

   Or

   For NIS server, map the Data Mover with the NIS server:

   ```
   $ server_nis server_<x> <IP address of NIS server>
   ```

   where:

   ‹x› = number of the Data Mover
3. Restart the Data Mover.
4. Export a file system for the users to access. Export a file system on page 33 provides more information.
5. On the PC, launch the PC-NFS or NFS Maestro software.
6. Type the username and password as required by the software package.

   The username and password are sent to the Data Mover that runs pcnfsd, which returns the UID or GID numbers for the PC client.

Hummingbird PC NFS client issues

Interoperability issues are identified in environments with both CIFS and Hummingbird PC NFS clients that access Microsoft Word or Corel WordPerfect files:

- Range locking — Normally, if a CIFS client opens a Microsoft Word file, and a Hummingbird (or any PC-NFS) client tries to delete the file, the delete request is refused due to the deny delete lock imposed when the CIFS client opened the file.

  The file is also range locked, however, with the offset not congruent with the start or end of the file. Thus, if a portion of the file is outside the range specified by range lock, the file can be written to. Write requests that fall within the range lock are denied. Users might experience unpredictable results because there is no method to determine which portions of the file are range locked.
- Share authentication — Hummingbird users have the option of overriding share authentication for VNX (or any other) drives when these drives are mounted on the client. In these cases, CIFS and Hummingbird clients have concurrent access to the file.
- Directory locking — If a CIFS client has a directory open in Windows Explorer and a Hummingbird client subsequently opens and attempts to close a WordPerfect file it has opened in the same directory, the Hummingbird client attempts to lock the file.

Because the file is opened by Windows Explorer, a lock exists and the Hummingbird lock request is denied. The Hummingbird client continuously issues lock requests to the server until the CIFS client closes the directory and releases its lock.

APPENDIX D

Parallel NFS on VNX for Block

This appendix describes how to configure and manage Parallel NFS (pNFS) on an EMC VNX for Block system.

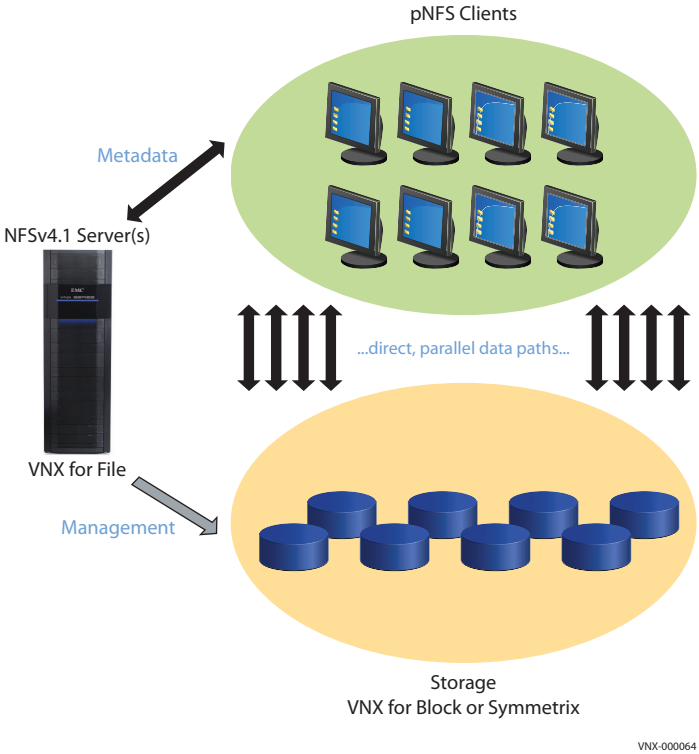
Topic included are:

- About pNFS 106
- Configuring pNFS for server 107
- Set up pNFS client 109
- Configuring SAN and storage 110

About pNFS

The pNFS is a part of NFS version 4.1 that allows clients to access storage devices directly and in parallel. The pNFS service starts automatically once the NFSv4 starts. NFSv4.1 facilitates to correct weaknesses and oversights, adding clarity and specificity by adding certain features and by supporting parallel access to files. The VNX provides a conforming implementation of NFSv4.1 in order to support pNFS. The pNFS architecture eliminates the scalability and performance issues associated with NFS servers in deployment today. This is achieved by the separation of data and metadata, and moving the metadata server out of the data path as shown in Parallel NFS on page 106.

Figure 3 Parallel NFS



Note

The implementation of NFSv4.1/pNFS provides support for IPv6 networks (including the iSCSI protocol).

In order to support pNFS, the VNX fully supports all mandatory NFSv4.1 functionality, including some of the smaller extensions and corrections from RFC 3530.

With pNFS, data access can be done using different protocols. The protocol that is used to access the data is known as “layout type”. NFS 4.1 related RFC allows for three possible pNFS layout types: file, block, and object. VNX supports block layout.

Layouts are recallable objects that assure the holder that direct access to the file data may be performed directly by the client and that no change to the location of the data that is inconsistent with that access should be made as long as the layout is held.

Mandatory features of NFSv4.1

NFSv4.1 introduces two new mandatory features that diverge from the NFSv4.0 specification.

The session model

The NFSv4.1 protocol requires sessions for all operations except for those that are used to initiate sessions. It introduces the SEQUENCE operation, which carries the session id, as well as other parameters related to session management.

A given client host can have multiple NFSv4.1 sessions. A session can have either one or two communication channels: a fore channel for issuing protocol requests and optionally, a back channel for receiving callbacks.

A channel can have zero or more transport connections associated with it. By binding multiple connections to a channel, the client can achieve trunking. In spite of having multiple sessions, the state information such as reservations, locks, delegations, and layouts are tied to the client and not to any particular session. Thus callbacks can be issued over back channels belonging to any session.

A client can end a session when idle in order to avoid using server resources unnecessarily.

Reclaiming state

With NFSv4.1 the clients may call an operation called RECLAIM_COMPLETE when they have reclaimed all the locks, possibly none that they wish to reclaim and extend the reclaim model to file system migrations. Such an operation affects servers that migrate a file system.

If the server maintains a list of lock-holding clients (but not their full lock state) in stable storage, then as soon as all such clients have declared their reclaim operations complete, it can end the grace period. The NFSv4.1 server allows clients to reclaim locks during the grace period only, and deny any OPEN operation other than those reclaiming a STATE during this period.

Configuring pNFS for server

The tasks to configure pNFS are:

- Set up pNFS service on page 107
- Stop pNFS on page 108
- Restart pNFS on page 108
- Mount the file system for the pNFS service on page 109
- Export a file system for pNFS access on page 109

Set up pNFS service

The pNFS service starts automatically when your server is configured for NFSv4. To configure your server for NFSv4:

Procedure

1. Ensure that the `/nas/server/slot_<x>/config` file where ‹x› is the slot number of the Data Mover that has the hivers set to 4. The nfs entry in the `/nas/server/slot_<x>/config` file should appear similar as follows:

   ```
   nfs config hivers=4
   ```

2. If the `hivers=4` is missing, add it manually at the end of the nfs config line.
3. Restart the Data Mover to bring the hivers change into effect by using this command syntax:

   ```
   $ server_cpu <movername> -reboot -monitor now
   ```

   where:

‹movername› = name of the Data Mover

4. To start the NFSv4 service, use this command syntax:

   ```
   $ server_nfs {<movername> | ALL} -v4 -service -start
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To start the NFSv4 service on server_2, type:

   ```
   $ server_nfs server_2 -v4 -service -start
   ```

   Output:

   ```
   server_2: done
   ```

Stop pNFS

Use this command to stop the pNFS service temporarily.

Note

When the pNFS service is stopped, all granted layouts are recalled and requests for new layouts is denied.

Procedure

1. To stop the pNFS service, use this command syntax:

   ```
   $ server_nfs {<movername> | ALL} -pnfs -service -stop
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To stop the pNFS service on server_2, type:

   ```
   $ server_nfs server_2 -pnfs -service -stop
   ```

   Output:

   ```
   server_2: done
   ```

Restart pNFS

Use this command to restart the pNFS service, if you have stopped it temporarily. Make sure that the NFSv4 service is running.

Procedure

1. To restart the pNFS service, use this command syntax:

   ```
   $ server_nfs {<movername> | ALL} -pnfs -service -start
   ```

   where:

   ‹movername› = name of the Data Mover

   Example:

   To restart the pNFS service on server_2, type:

   ```
   $ server_nfs server_2 -pnfs -service -start
   ```

   Output:

```
server_2: done
```

Mount the file system for the pNFS service

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

Procedure

1. To mount the file system on a Data Mover for pNFS service, use this command syntax:

   ```
   $ server_mount <movername> -option pnfs <fs_name>
   /<mount_point>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹fs_name› = file system to be mounted

   ‹mount_point› = path to mount point for the Data Mover

   Example:

   To mount a file system on server_2 using a pnfs option, type:

   $ server_mount server_2 -option pnfs s2fs1 /s2mt1

   Output:

   ```
   server_2 : done
   ```

Export a file system for pNFS access

Use the server_export command to export a file system for pNFS access. By default, the server_export command exports the file system as read-write for all NFS clients and uses UNIX user authentication.

Procedure

1. To export a file system for pNFS access, use this command syntax:

   ```
   $ server_export <movername> -Protocol nfs <fs_name>
   /<mount_point>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹fs_name› = file system to be mounted

   ‹mount_point› = path to mount point for the Data Mover

   Example:

   To export a file system for pNFS access, type:

   $ server_export server_2 -Protocol nfs s2fs1 /s2mt1

   Output:

   ```
   server_2: done
   ```

Set up pNFS client

Before you mount the pNFS file system, make sure that the pNFS client is correctly installed:

Procedure

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

   ```
   # mount -type nfs4 -option minorversion=1
   <movername>:/<fs_name> /<mount_point>
   ```

   where:

   ‹movername› = name of the Data Mover

   ‹fs_name› = file system to be mounted

   ‹mount_point› = path to mount point for the Data Mover

   Note

   The mount is successful if you do not receive any error messages.

   Example:

   To mount a pNFS file system on server_2, type:

   ```
   # mount -type nfs4 -option minorversion=1 server_2:/fs1_pnfs
   /mnt1_pnfs
   ```

2. To verify if the mount was created, run the following command:

   ```
   # grep LAYOUT /proc/self/mountstats
   ```

   Output:

   ```
   PNFS_LAYOUTGET: 0 0 0 0 0 0 0 0
   PNFS_LAYOUTCOMMIT: 0 0 0 0 0 0 0 0
   PNFS_LAYOUTRETURN: 0 0 0 0 0 0 0 0
   ```

   Note

   If the output has non-zero values, refer to /var/log/messages for error messages.

3. If you are using iSCSI, you need to set up iSCSI on the client. Procedure varies according to the operating system you are using. As a result, the iSCSI initiator should be created with an iSCSI Qualified Name (IQN).

   If you are using Fibre Channel, you need to set up a host bus adapter (HBA).

Configuring SAN and storage

This section describes how to configure the SAN switch along with specific configuration information for VNX for block. You may also have EMC Symmetrix® as the storage system. For more information on how to configure EMC Symmetrix® for pNFS, go to the EMC Online Support website at http://Support.EMC.com and search for related technical documentation.

Set up storage groups for pNFS client by using the Unipshere for block.

Register iSCSI hosts that you created while setting up the pNFS client by using Unisphere for block.

Install the Fibre Channel switch (FC configuration)

To set up the Fibre Channel switch, complete these tasks:

Note

Configure zoning as single initiator, meaning that each HBA port will have its own zone in which it is the only HBA port.

Procedure

1. Install the Fibre Channel switch (if not already installed).
2. Connect cables from each HBA Fibre Channel port to a switch port.
3. Verify the HBA connection to the switch by checking the LEDs for the switch port connected to the HBA port.

Zone the SAN switch (FC configuration)

To configure and zone a Fibre Channel switch:

Note

- For more information, refer to the documentation that ships with the Fibre Channel switch.
- Configure the VNX for block so that each target is zoned to an SP A and SP B port. Configure the Symmetrix so that it is zoned to a single Fibre Channel Director.

Procedure

1. Record all attached port World Wide Names (WWNs).
2. Log in to the iSCSI-to-Fibre Channel bridge or MDS Fibre Channel switch console by using CLI commands or the Fabric Manager.
3. Create a zone for each Fibre Channel HBA port and its associated Fibre Channel Target.

Configure the iSCSI-to-Fibre-Channel bridge (iSCSI configuration)

This section presents an overview of configuring and zoning the iSCSI-to-Fibre Channel bridge for an iSCSI configuration. Use your site-specific parameters for these steps.

When configuring the iSCSI-to-Fibre Channel bridge for an iSCSI configuration:

- The EMC VNX pNFS over iSCSI configuration does not use the EMC VNX iSCSI target feature. It relies on the iSCSI initiator and an IP-SAN compatible switch as the iSCSI target to bridge traffic to Fibre Channel.
- Verify the IP-SAN switch requirements to identify supported switch models.

The pNFS environment requires a Fibre Channel switch capable of iSCSI-to-Fibre Channel bridging. The following switch characteristics are required:

- An iSCSI-to-Fibre Channel bridge that supports the pNFS architecture.
- pNFS IP-SAN switches require specific firmware revisions for proper operation.
- The SAN OS version with the E-Lab Interoperability Navigator for supported versions.

Install one of the following iSCSI IPS (IP-SAN) modules:

- 8GE IPS blade
- 14FC/2/GE multi protocol services module
- 18FC/4/GE multi protocol services module

Note

The E-Lab Interoperability Navigator contains definitive information on supported software and hardware for VNX network-attached storage (NAS) products.

Enable iSCSI on the MDS console (iSCSI configuration)

To enable iSCSI on the MDS console:

Procedure

1. Enable iSCSI and add the iSCSI interface to the VSAN:

```
switch # config t
switch(config)# iscsi enable
switch(config)# iscsi interface vsan-membership
```

2. Import the Fibre Channel targets so the MDS console can communicate with the iSCSI initiator and set the authentication:

```
switch(config)# iscsi import target fc
switch(config)# iscsi authentication none
```

Configure the MDS iSCSI port for Gigabit Ethernet (iSCSI configuration)

Configure the MDS iSCSI port as a normal Gigabit Ethernet port:

```
switch(config)# interface GigabitEthernet2/1
switch(config-if)# <port ip address> <subnet mask>
switch(config-if)# iscsi authentication none
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# ip routing
switch(config)#
```

Note

The iSCSI-to-Fibre Channel bridge can also be configured and managed through the Fabric Manager GUI.

Configure the iSCSI port proxy and TCP parameters (iSCSI configuration)

Configure the iSCSI-to-Fibre Channel bridge to act as a proxy initiator so that the VNX for block does not need to be configured with the WWNs for all the initiators:

Procedure

1. Set the iSCSI-to-Fibre Channel bridge to act as a proxy initiator:

```
switch(config)# interface iscsi2/1
switch(config-if)# switchport proxy-initiator
switch(config-if)#
```

2. Set the TCP parameters:

```
switch(config-if)# tcp send-buffer-size 16384
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# exit
switch #
```

3. Retrieve the system-assigned WWNs by typing:

```
switch # show interface iscsi2/1
```

Output:

```
Hardware is GigabitEthernet
      Port WWN is 21:d9:00:0d:ec:01:5f:40
```

```
        Admin port mode is ISCSI
        Port vsan is 2
        iSCSI initiator is identified by name
        Number of iSCSI session: 0 (discovery session: 0)
        Number of TCP connection: 0
        Configured TCP parameters
          Local Port is 3260
          PMTU discover is enabled, reset timeout is 3600 sec
          Keepalive-timeout is 60 sec
          Minimum-retransmit-time is 300 ms
          Max-retransmissions 4
          Sack is enabled
          QOS code point is 0
          Maximum allowed bandwidth is 1000000 kbps
          Minimum available bandwidth is 70000 kbps
          Estimated round trip time is 1000 usec
          Send buffer size is 16384 KB
          Congestion window monitoring is enabled, burst size is 50
KB
          Configured maximum jitter is 500 us
Forwarding mode: store-and-forward
  TMF Queueing Mode: enabled
  Proxy Initiator Mode: enabled
      nWWN is 20:10:00:0d:ec:01:5f:42 (system-assigned)
      pWWN is 20:11:00:0d:ec:01:5f:42 (system-assigned)
  5 minutes input rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
  5 minutes output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
  iSCSI statistics
    Input 0 packets, 0 bytes
      Command 0 pdus, Data-out 0 pdus, 0 bytes, Unsolicited 0
bytes
    Output 0 packets, 0 bytes
      Response 0 pdus (with sense 0), R2T 0 pdus
      Data-in 0 pdus, 0 bytes
```

Create and configure a new zone for iSCSI proxy (iSCSI configuration)

Before creating a new zone for the iSCSI proxy, check the FC ports connected to SP A and SP B of the system. In the example, SP A is connected to port FC4/2 and SP B is connected to port FC4/4.

In principle, two ports may be zoned with each proxy. However, the throughput of iSCSI will still be 1 Gb. One port must be connected to each SP for high availability.

Using Navicli or Unisphere, obtain the pWWN for SP A and SP B. Create the new zone named using the name of the iSCSI port. PROXY2-1 is used in the example:

```
switch # config t
switch(config)# zone name PROXY2-1 vsan 2
switch(config-zone)# member pwwn 50:06:01:60:10:60:20:35
switch(config-zone)# member pwwn 50:06:01:60:10:60:20:3e
switch(config-zone)# member pwwn 20:11:00:0d:ec:01:5f:42
switch(config-zone)# exit
```

Activate the zone (iSCSI configuration)

After creating the zone, activate it for the switch to know it exists.

To include and activate the newly created zone into the active zones, set:

```
switch(config)# zoneset name PROXIES vsan 2
switch(config-zoneset)# member PROXY2-1
switch(config-zoneset)# exit
switch(config)# zoneset activate name PROXIES vsan 2
```

Note

The vendor's documentation contains specific information on these switches.

Enable SP A and SP B FC interfaces (FC configuration)

Enable the Fibre Channel ports for the storage processors. Type:

```
switch(config)# interface fc4/2
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface fc4/4
switch(config-if)# no shutdown
switch(config-if)# exit
```

Add a VSAN for iSCSI/FC and enabling zones

Use the following example to add the iSCSI blade in slot 2, port 1 and Fibre Channel (FC) blade in slot 4, ports 2 and 4 to MDS VSAN 2. Then, enable the default zone to permit VSANs 1-4093:

Procedure

1. Add a VSAN for iSCSI and Fibre Channel to the iSCSI-to-Fibre Channel bridge:

   ```
   switch(config)# show vsan database
       switch(config-vsan-db)# vsan 2
       switch(config-vsan-db)# vsan 2 interface iscsi2/1
       switch(config-vsan-db)# vsan 2 interface fc4/4
       switch(config-vsan-db)# vsan 2 interface fc4/2
       switch(config-vsan-db)# exit
   ```

2. Enable zones on the iSCSI-to-Fibre Channel bridge:

   ```
   switch(config)# zone default-zone permit vsan 1-4093
   ```

3. Log off the iSCSI-to-Fibre Channel bridge or MDS Fibre Channel switch console:

   ```
   switch(config)# exit
   ```

Create a security file on a VNX for File

A system will not accept a Secure CLI command unless the user who issues the command has a valid user account on the storage system. You can configure a Unisphere 6.X security file to issue Secure CLI commands on the server. Secure CLI commands require the servers (or the password prompt) in each command line; you do not need to provide these in the command line if you create a security file.

To create a security file:

Procedure

1. Log in to the Control Station as NAS administrator.
2. If you use the same settings for the username and passwords, as done on the switch, type:

   ```
   /nas/sbin/naviseccli -h <hostname:IP address> security
   -AddUserSecurity -scope 0 -user nasadmin -password nasadmin
   ```

   Note

   Specify the hostname as SP A or SP B.

3. If you are using a new VNX for block CX3 or CX4 series system with NAS v5.5 and content is running in non-secure mode, type:

   ```
   /nas/sbin/naviseccli -h <hostname:IP address> security
   -AddUserSecurity -scope 0 -user emc -password emc
   ```

VNX for block iSCSI port configuration

This section describes how to set up the VNX for block for an iSCSI configuration:

Note

The IP addresses of all systems ‹hostname:IP address› can be found in the /etc/hosts file on the Control Station. If multiple systems are used, EMC recommends registering them in the /etc/hosts file.

Procedure

1. Use the following naviseccli command to configure iSCSI target hostname SP A, port IP address 0 on the storage array:

```
/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp a -portid 0 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

It is recommended that you consult with your Network
Manager to determine the correct settings before applying
these changes. Changing the port properties may disrupt iSCSI
traffic to all ports on this SP. Initiator configuration changes
may be necessary to regain connections. Do you really want to
perform this action (y/n)? y
```

2. Continue for SP A ports 1 to 3 and SP B ports 0 to 3:

```
/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp a -portid 1 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp a -portid 2 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp a -portid 3 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp b -portid 0 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp b -portid 1 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp b -portid 2 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>

/nas/sbin/naviseccli -h <hostname:IP address> connection
-setport -sp b -portid 3 -address <port IP address>
-subnetmask <subnet mask> -gateway <gateway IP address>
```

Access Logix configuration

This section describes how to set up an EMC Access Logix™ configuration, create storage groups, add LUNs, and set failovermode and the arraycommpath.

Setting failovermode and the arraycommpath by using Access Logix

The naviseccli failovermode command enables or disables the type of trespass needed for the failover software. This method of setting -failovermode works for storage systems with Access Logix only.

To discover the current setting of failovermode, use the port-list-failovermode or port-list-all command:

⚠ CAUTION

Changing the failovermode setting may force the system to reboot. Changing the failovermode to the wrong value will make the storage group inaccessible to any connected server.

1. To set failovermode to 1 (unified storage only), type:

```
/nas/sbin/naviseccli -h <hostname:IP address>
failovermode 1

WARNING: Previous Failovermode setting will be lost!
DO YOU WISH TO CONTINUE (y/n)? y
```

2. To set arraycommpath to 1 (unified storage only), type:

```
/nas/sbin/naviseccli -h <hostname:IP address> arraycommpath 1

WARNING: Previous arraycommpath setting will be lost!
DO YOU WISH TO CONTINUE (y/n)? y
```

These commands enable failovermode and arraycommpath for the system.

Creating storage groups and adding LUNs

This section describes how to create storage groups, add LUNs to the storage groups, and configure the storage groups.

The IP addresses of all systems ‹hostname:IP address› can be found in the /etc/hosts file on the Control Station. If multiple systems are used, EMC recommends registering them in the /etc/hosts file:

Note

Specify the hostname as SP A or SP B.

1. Specify the hostname as SP A or SP B:

```
$ /nas/sbin/navicli -h <hostname:IP address> storagegroup
-create -gname PNFS_Clients
```

2. Add LUNs to the storage group:

```
$ /nas/sbin/navicli -h <hostname:IP address> storagegroup
-addhlu -gname PNFS_Clients -hlu 0 -alu 16
```

3. Continue for all LUNs:

```
$ /nas/sbin/navicli -h <hostname:IP address> storagegroup
-addhlu -gname PNFS_Clients -hlu 1 -alu 17
```

INDEX

32bitClient parameter 77

A

access control lists (ACLs) 21
access, disabling NFS 67
acl.sortAces parameter 49
authentication
    for PC clients 19
    for UNIX and Linux clients 19
automounter feature 68

C

CDMS 10
CIFS
    parameters
        acl.sortAces 49
client contexts 24

D

delegation 22
disabling NFS access 67
DNS
    configuring for secure NFS 50
    parameters
        updatePTRrecord 43

E

EMC E-Lab Navigator 84
error messages 88
exporting
    using Kerberos authentication 39, 40
    using UNIX and Kerberos authentication 40
    using UNIX authentication 39

F

file locking 22
file systems
    exporting with Kerberos authentication 39, 40
    exporting with UNIX and Kerberos authentication 40
    exporting with UNIX authentication 39
    mounting 31

H

hivers option 26

I

internationalization 21

K

Kerberos
    using a UNIX or Linux KDC with secure NFS 50
    using a Windows KDC with secure NFS 42

L

Linux Kerberos KDC, configure for secure NFS 50

M

managing 66
map file
    copying 68
    creating 68
messages, error 88
mode bits 20
mounting
    file systems 31
multihomed Data Movers 54

N

naming services 28
nas_automountmap 68
netd file 26
NFS
    file locking 22
    authentication for PC clients 19
    creating a mount point 30
    defined 9
    disabling access 67
    displaying statistics 85
    exporting a file system 34
    file locking 22
    internationalization 21
    managing 66
    managing secure NFS 77
    mounting a file system 31
NFSv4
    access control lists 21
    client contexts 24
    creating a mount point 30
    delegation 22
    displaying statistics 85
    error messages 88
    exporting a file system 33, 34
    managing 72
    managing secure NFS 77
    mounting a file system 32
    naming services 28
    parameters
        32bitClient 77
    recallTimeout 72
    setup procedures 26, 33
    troubleshooting 84
    Unicode 29
NIS
    configuring for secure NFS 42, 50

P

parameters
- 32bitClient 77
- acl.sortAces 49
- recallTimeout 72
- secureExportMode 95
- updatePTRrecord 43

preventing NFS setuid and setgid bit access 36
principals
- adding service principals 55
- mapping to UIDs 57

protocol
- NFS 9

R

recallTimeout parameter 72
reexporting 67
reexporting file systems 67

S

secure NFS
- using a UNIX or Linux Kerberos KDC 50
- using a Windows Kerberos KDC 42

secureExportMode parameter 95
security
- Kerberos based 19
- UNIX based 19

server_export
- access options 35
- Kerberos 39
- Kerberos security 39
- NFSv4
  - access by NFSv4 clients only 39
- setting access levels 35
- UNIX security 39
- VLAN access 38

server_kerberos
- creating realm 51
- creating service principals 55

server_mountpoint 30
server_mountt 31
server_name 51
server_nfs
- deleting principals 80
- displaying mapping entry 79
- displaying NFSv4 clients 75
- displaying NFSv4 status 84
- displaying user attributes 79
- mapping Kerberos user principal names to UIDs 48
- releasing authentication 80
- releasing NFSv4 clients 76
- setting the secure NFS service instance 46, 53
- starting NFSv4 74
- stopping NFSv4 74

server_umount 33
service principals
- adding using the KDC 56

Solaris Kerberos KDC, configuring for secure NFSLinux
  - Kerberos KDC, configuring for secure NFS 50

system access 18
system access control 18

T

time services, configuring for secure NFS 42, 50
troubleshooting 83

U

UIDs, mapping to user principal names 57
unexporting 67
unexporting, file systems 66
Unicode 29
UNIX security 19
unmounting 33
updatePTRrecord parameter 43
user authentication 19
user mapping 48
UTF-8 21

V

VLAN, specifying access by 38
VNX FileMover feature 10