



EMC® Celerra® Network Server

Release 5.6.46

Configuring NFS on EMC Celerra

P/N 300-004-152

REV A03

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Copyright © 1998 - 2009 EMC Corporation. All rights reserved.

Published August 2009

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.

For the most up-to-date regulatory document for your product line, go to the Technical Documentation and Advisories section on EMC Powerlink.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Preface	7
Chapter 1: Introduction	11
System requirements.....	12
Restrictions.....	13
User interface choices.....	14
Related information.....	14
Chapter 2: Concepts	17
Overview.....	18
Planning considerations.....	19
System access control.....	19
User authentication.....	19
User access control by using mode bits.....	21
User access control using ACLs.....	21
Internationalization.....	22
NFSv4 domain name.....	22
NFSv4 delegation.....	23
File locking.....	24
Client contexts.....	25
Limiting access to NFSv4 clients only.....	25
Chapter 3: Configuring	27
Getting started.....	28
Enable NFSv4.....	28

Configure the NFSv4 domain.....	28
Configure access to naming services.....	30
Convert file systems to Unicode.....	31
Create a mount point.....	32
List the mount point.....	32
Mount a file system.....	33
Mount a file system for basic NFS access.....	33
Mount a file system for NFSv4 access.....	34
Unmount a file system.....	35
Export a file system.....	36
Export a file system for basic NFS access.....	37
Levels of access.....	38
Multiple levels of a file system.....	39
Restrict access by VLAN.....	40
Specify NFSv4 access.....	41
Specify the user authentication method.....	42
Configure secure NFS by using a Windows Kerberos KDC.....	43
Enable dynamic updates of reverse lookups.....	44
Configure CIFS.....	46
Set the secure NFS service instance.....	49
Set the NFS user and group mapping service.....	52
Reorder ACEs (multiprotocol environment only).....	53
Configure secure NFS using a UNIX or Linux Kerberos KDC.....	54
Set the Data Mover name.....	55
Configure the Kerberos realm.....	56
Set the secure NFS service instance.....	57
Create the NFS Kerberos service principals.....	61
Map user principal names to UIDs.....	64
Use the Solaris mapping utility gsscred.....	66
Copy the mapping file.....	66
Copy the mapping file to an NIS server	67
Copy the mapping file to a Data Mover	68
Create a local mapping file.....	69
Mount a Celerra file system on a client system.....	71
Chapter 4: Managing.....	73
Manage NFS.....	74

Unexport the NFS path to a Celerra file system.....	74
Reexport all NFS paths on a Celerra Network Server.....	75
Disable NFS access to all file systems on a Data Mover.....	75
NFS automounter feature support.....	76
Hide NFS exports.....	80
Manage NFSv4.....	82
Modify state duration.....	82
Change the delegation recall timeout.....	83
Modify number of usable nodes.....	83
Display NFSv4 service status.....	84
Stop the NFSv4 service.....	84
Restart the NFSv4 service.....	85
Display NFSv4 clients.....	85
Display information about NFSv4 clients.....	86
Release NFSv4 clients.....	88
Support 32-bit and 64-bit NFS clients.....	88
Manage Secure NFS.....	89
View keytab entries.....	89
Display all secure NFS service instances.....	90
Display user attributes.....	91
Display information about a user in a local mapping file.....	92
Delete a user from a local mapping file.....	93
Delete service principals.....	93
Release authentication.....	94
Chapter 5: Troubleshooting.....	95
EMC E-Lab Interoperability Navigator.....	96
Troubleshooting NFSv4.....	96
Display NFSv4 status.....	96
Display NFS statistics.....	97
Troubleshooting secure NFS.....	102
Cannot access file system.....	103
Release user authentication.....	103
Error messages.....	105
NFSv4 error messages.....	105
EMC Training and Professional Services.....	105

Appendix A: System Access Behavior.....107
 Behavior when specifying combinations of access modes.....108
 Rules for resolving conflicts among hosts, subnets, and netgroups.....109
 Specify read-only as the default access mode.....111

Appendix B: User Authentication Behavior.....113
 General rules when specifying security options.....114
 Root access mode.....115

Appendix C: NFS Authentication Daemon for PC Clients.....117
 PC client access.....118
 Set up PC client software.....119
 Hummingbird PC NFS client issues.....119

Terminology.....121

Index.....123

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

EMC uses the following conventions for special notices:



A caution contains information essential to avoid data loss or damage to the system or equipment.

Important: An important note contains information essential to operation of the software.

Note: A note presents information that is important, but not hazard-related.

Hint: A note that provides suggested advice to users, often involving follow-on activity for a particular action.

Typographical conventions

EMC uses the following type style conventions in this document.

Type style	Used for
Normal	<ul style="list-style-type: none"> ◆ Running text ◆ Names of resources, attributes, pools, clauses, functions, and utilities
<i>Italic</i>	<ul style="list-style-type: none"> ◆ Full titles of publications (citations) ◆ Variables, in running text
Helvetica bold	<ul style="list-style-type: none"> ◆ User interface elements (what users specifically select, click, or press) ◆ Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) ◆ Command and program options
Courier	URLs, email addresses, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, command names (in running text), user input (such as commands), and notifications (system output, system messages, etc.)
Courier bold	Command names, in syntax diagrams
<i>Courier italic</i>	Variables, in syntax diagrams (except Celerra) and user input
<>	Variables, in Celerra syntax diagrams
[]	Optional selections
{}	Required selections
	Alternative selections. The bar means "or"
...	Nonessential information omitted from an example

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 the EMC Powerlink website (registration required) at <http://Powerlink.EMC.com>.

Technical support — For technical support, go to Powerlink and select **Support**. On the Support page, you can access Support Forums, request a product enhancement, talk directly to an EMC representative, or open a service request. To open a service request, you must have a valid support agreement. Please contact your EMC Sales Representative for details about obtaining a valid support agreement or to answer any questions about your account.

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`

The EMC[®] Celerra[®] Network Server 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, Celerra provides access to Celerra file systems for clients that use versions 2, 3, and 4 of the NFS protocol.

This document is part of the Celerra Network Server information set and is intended for system administrators responsible for configuring and managing NFS on the Celerra Network Server.

- ◆ [System requirements on page 12](#)
- ◆ [Restrictions on page 13](#)
- ◆ [User interface choices on page 14](#)
- ◆ [Related information on page 14](#)

System requirements

Table 1 on page 12 describes the Celerra Network Server software, hardware, network, and storage configurations required for configuring NFS as described in this document.

Table 1. System requirements

Software	<p>Celerra Network Server version 5.6.46</p> <p>For secure NFS using UNIX or Linux-based Kerberos:</p> <p>Sun Enterprise Authentication Mechanism (SEAM) software or Linux KDC that runs Kerberos Version 5</p> <hr/> <p>Note: KDCs from other UNIX systems have not been tested.</p> <hr/> <p>For secure NFS using Windows-based Kerberos: Windows 2000 or Windows Server 2003 domain.</p> <p>For secure NFS, the client computer must be running:</p> <ul style="list-style-type: none"> ◆ 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 <hr/> <p>Note: Other clients have not been tested.</p> <hr/> <ul style="list-style-type: none"> ◆ DNS ◆ NTP (Network Time Protocol) server <p>Windows environments require that you configure Celerra in the Active Directory.</p>
Hardware	No specific requirements
Network	No specific requirements
Storage	No specific requirements

Restrictions

The Celerra Network Server supports all NFS features with the following restrictions:

- ◆ A Data Mover can host only one instance of an NFS server and domain.
- ◆ Secure NFS is supported only for NFSv3 and NFSv4.
- ◆ If you are using the Celerra FileMover feature or CDMS, the NFS communication between the Celerra Network Server and secondary storage requires UNIX AUTH_SYS authentication and does not support NFSv4.
- ◆ The MPFS block data access channel is only used if the MPFS client is using NFSv3 or CIFS to access Celerra.
- ◆ Celerra does not support creating NFS exports on VDMs. However, NFS clients can access the file systems mounted on VDM by creating NFS exports on the Data Mover in which the VDM is configured. The following restrictions apply:
 - The NFS exports created this way are not part of the VDM configuration and are not carried over if the VDM is moved to another physical Data Mover or failed over to a remote Celerra in case of replication.
 - NFS export of a file system mounted on VDM can be done through the CLI `server_export` and not with Celerra Manager.
 - NFS export cannot be restricted to a specific VDM. However, the `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 Celerra when you create the VDM. The NFS export aliasing can be used to change the view of NFS exports to the client.

The Celerra implementation of NFSv4 supports:

- ◆ NFSv4 over the networking protocol IPv4.
- ◆ UNIX and Kerberos v5 for user authentication.
- ◆ All the defined file attributes except the recommended attributes `archive` and `fs_location`.
- ◆ UTF-8 and ASCII.
- ◆ Pseudo-root file systems.
- ◆ Access control lists.
- ◆ Nested mount file systems.

The Celerra implementation of NFSv4 does not support:

- ◆ Blocking locks feature.
- ◆ Access control that uses Windows SIDs.

- ◆ Public key-based authentication (SPKM-3 and Lipkey).
- ◆ ASCII code pages.
- ◆ Log messages that indicate NFSv4 service status (starting and stopping) written to the Data Mover's server log.

User interface choices

The Celerra Network Server 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 Celerra Manager. 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 Celerra:

- ◆ *Learning about EMC Celerra on the EMC Celerra Network Server Documentation CD*
- ◆ Celerra Manager online help
- ◆ *Monitoring EMC Celerra*

Limitations

You cannot use the Celerra Manager to configure the following features:

- ◆ Automounter
- ◆ Secure NFS
- ◆ NFSv4 features

The *EMC Celerra Network Server Release Notes* contain additional, late-breaking information about Celerra management applications.

Related information

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

- ◆ *EMC Celerra Glossary*
- ◆ *Configuring and Managing CIFS on EMC Celerra*
- ◆ *Managing EMC Celerra for a Multiprotocol Environment*
- ◆ *Using International Character Sets with EMC Celerra*
- ◆ *Configuring EMC Celerra Naming Services*
- ◆ *Configuring EMC Celerra Time Services*
- ◆ *Managing EMC Celerra Volumes and File Systems Manually*

- ◆ *EMC Celerra Network Server Command Reference Manual*
- ◆ Online Celerra man pages
- ◆ *EMC Celerra Network Server Parameters Guide*

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.

The *EMC Celerra Network Server Documentation CD* supplied with the Celerra and also available on [Powerlink](#)[®] provides the complete set of EMC Celerra customer publications. After logging in to Powerlink, go to **Support > Technical Documentation and Advisories > Hardware/Platforms Documentation > Celerra Network Server**. On this page, click Add to Favorites. The Favorites section on your Powerlink home page provides a link that takes you directly to this page.

Celerra support demos

Celerra Support Demos are available on Powerlink. Use these instructional videos to learn how to perform a variety of Celerra configuration and management tasks. After logging in to Powerlink, go to **Support > Product and Diagnostic Tools > Celerra Tools > Celerra Support Demos**.

The concepts and planning considerations to understand NFS are:

- ◆ [Overview on page 18](#)
- ◆ [Planning considerations on page 19](#)

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 Celerra Network Server 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.

Figure 1 on page 18 shows a Data Mover configured for NFS clients.

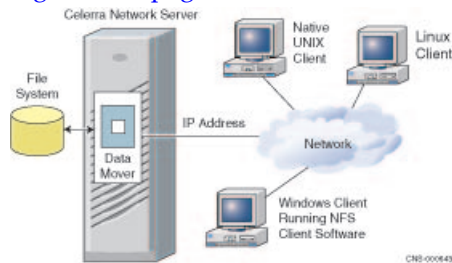


Figure 1. NFS File System Configuration

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 requesting information with the access rights defined for the exported file system.
 - Performing user access control on the file system object.

The Celerra Network Server 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 EMC Celerra for a Multiprotocol Environment provides information about configuring the Celerra Network Server to support both NFS and CIFS.

Configuring and Managing CIFS on EMC Celerra provides information about CIFS only.

Planning considerations

Based on the NFS environment requirements, review the following considerations before configuring NFS on the Celerra Network Server:

- ◆ [System access control on page 19](#)
- ◆ [User authentication on page 19](#)
- ◆ [User access control using mode bits on page 21](#)
- ◆ [User access control using ACLs on page 21](#)
- ◆ [Internationalization on page 22](#)
- ◆ [NFSv4 domain name on page 22](#)
- ◆ [NFSv4 delegation on page 23](#)
- ◆ [File locking on page 24](#)
- ◆ [Client contexts on page 25](#)
- ◆ [Limiting access to NFSv4 clients only on page 25](#)

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 37](#) describes this procedure. [Appendix A](#) describes how to interpret the read and write access you set for NFS clients of exported file systems.

User authentication

User authentication enables a Celerra 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 Celerra Network Server's NFS service authenticates users through the mechanisms listed in [Table 2 on page 20](#).

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 <code>security.maxgroups</code> allows you to increase the number of UNIX groups for NFS clients from the default limit of 16 to 128. The <i>EMC Celerra Network Server Parameters Guide</i> 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 user's Kerberos principal is mapped to the UID by using a mapping service.
Authentication daemon	For a Windows system (PC client) that uses NFS to access the Celerra Network Server, an authentication daemon, typically <code>rpc.pcnfsd</code> or <code>pcnfsd</code> , 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 supports secure NFS by using either UNIX or Linux or Windows Kerberos Key Distribution Center (KDC).

[Specify the user authentication method on page 42](#) describes how to specify the user authentication method by using the `sec` option on the `server_export` command. [Appendix B](#) describes how to interpret the user authentication that was set for NFS clients of exported file systems.

[Appendix C](#) describes how to use the authentication daemon.

[Configure secure NFS by using a Windows Kerberos KDC on page 43](#) and [Configure secure NFS using a UNIX or Linux Kerberos KDC on page 54](#) describe the procedures for configuring secure NFS by using either Windows or UNIX or Linux Kerberos.

User access control by using mode bits

The Celerra Network Server performs mode-bit-based access control for NFS clients when the file system is mounted with one of the following Celerra 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, execute) triplets for each category of users (user or group or other), as shown in the following example of a file system directory:

```
lrwxrwxrwx 1 kcb eng 10 Dec 9 13:42 xyz.doc -> xyz.html
-rw-r--r-- 1 kcb eng 1862 Jan 2 14:32 abc.html
drwxr-xr-x 2 kcb eng 5096 Mar 9 11:30 schedule
```

User
Other
Group

CNS-000537

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, an `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 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 *EMC Celerra Network Server Parameters Guide* provides more information. When a user requests access to a file system object, the Celerra Network Server compares the user's credentials with the permissions on that object.

User access control 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 Celerra Network Server has a single implementation of ACLs, which provides consistent access control no matter which protocol (NFS or CIFS) is used to access data.

The Celerra Network Server performs ACL-based access control for NFS clients when the file system is mounted with one of the following Celerra access policies: NT, SECURE, MIXED, 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 34](#) describes this procedure. *Managing EMC Celerra for a Multiprotocol Environment* provides detailed information about configuring the Celerra 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. Regardless, access control to file system objects is not affected.

Internationalization

The Celerra Network Server supports clients in environments that use multibyte character sets. Multibyte character sets are supported by enabling universal character encoding standards (Unicode). The Celerra Network Server 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 Celerra character encoding method must be changed from the default to UTF-8.

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

NFSv4 domain name

Every Celerra Network Server user must be identified by a unique numeric UID and GID to indicate the ownership of directories and files. The Celerra Network Server 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 Celerra Network Server 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 Celerra Network Server, you must specify an NFSv4 domain name by using the `nfsv4.domain` parameter. [Configure the NFSv4 domain on page 28](#) 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 delegation

When using NFSv4, the Celerra Network Server 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 Celerra Network Server 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 34](#) describes how to specify the delegation level. [Manage NFSv4 on page 82](#) describes how to modify delegation operation by using NFSv4 parameters.

File locking

The Celerra Network Server 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. Celerra 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.

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 33](#) describes how to configure locks on a file system based on the Celerra Network Server.

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 Celerra Network Server 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 82](#) 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 Celerra Network Server 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 41](#) describes how to specify the NFSv4 option on the `server_export` command.

The tasks to configure NFS are:

- ◆ [Getting started on page 28](#)
- ◆ [Create a mount point on page 32](#)
- ◆ [Mount a file system on page 33](#)
- ◆ [Export a file system on page 36](#)
- ◆ [Configure secure NFS by using a Windows Kerberos KDC on page 43](#)
- ◆ [Configure secure NFS using a UNIX or Linux Kerberos KDC on page 54](#)
- ◆ [Use the Solaris mapping utility gsscred on page 66](#)
- ◆ [Mount a Celerra file system on a client system on page 71](#)

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 Celerra Network Server. However, if you are planning to use NFSv4, perform the following setup procedures:

- ◆ [Enable NFSv4 on page 28](#)
- ◆ [Configure the NFSv4 domain on page 28](#)
- ◆ [Configure access to naming services on page 30](#)
- ◆ [Convert file systems to Unicode on page 31](#)

Enable NFSv4

To enable NFSv4 support on the Celerra Network Server.

Step	Action
1.	With a text editor, open the file <code>/nas/server/slot_<x>/netd</code> , where <code><x></code> is the number of the Data Mover.
2.	On the <code>nfs start</code> line, append the option <code>hivers=4</code> or, if the option already appears, be sure the value is set to 4. Save and close the file. For example: <code>nfs start openfiles=240000 nfsd=256 hivers=4</code> Note: Do not change any other lines or options in the file.
3.	Reboot the Data Mover by using this command syntax: <code>server_cpu <movername> -reboot now</code> where: <code><movername></code> = 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. *EMC Celerra Network Server Parameters Guide* 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.

To configure the NFS domain:

- ◆ [Specify the NFSv4 domain name on page 29](#)
- ◆ [List the NFSv4 parameters on page 30](#)

Specify the NFSv4 domain name

Action
<p>To specify the NFSv4 domain name, use this command syntax:</p> <pre>\$ server_param <movername> -facility nfsv4 -modify domain -value <new_value></pre> <p>where:</p> <p><movername> = name of the specified Data Mover</p> <p><new_value> = name for the domain</p> <p>Example:</p> <p>To set the domain name to emc.com, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -modify domain -value emc.com</pre>
Output
<pre>server_2 : done</pre>

Verify the domain parameter

Action
<p>To verify whether the domain parameter was set, use this command syntax:</p> <pre>\$ server_param <movername> -facility nfsv4 -info domain</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To verify whether the domain parameter was set, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -info domain</pre>

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

Action

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
```

Configure access to naming services

Each Data Mover on a Celerra Network Server 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.

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 EMC Celerra Naming Services provides more information about configuring and managing naming services.

Note: The Celerra Network Server and all NFSv4 clients that access the Celerra 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

Convert file systems to Unicode

NFSv4 requires UTF-8 for character encoding. By default, a Celerra system is configured to use ASCII Latin-1 character encoding. Before accessing data through NFSv4 clients, you must change the character encoding method to UTF-8 in the `/nas/site/locale/xlt.cfg` file.

If you are installing a new Celerra Network Server, you can enable Unicode during the installation process before creating user files and directories. If you have an existing Celerra Network Server, 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 configuration files are installed.

Using International Character Sets with EMC Celerra describes how to configure and manage Unicode.



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

Create a mount point

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

Action
<p>To create a mount point for a file system, use this command syntax:</p> <pre>\$ server_mountpoint <movername> -create <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><pathname> = path to the mount point</p> <p>Example:</p> <p>To create a mount point called /usr1 on server_2, type:</p> <pre>\$ server_mountpoint server_2 -create /usr1</pre>
Output
<pre>server_2 : done</pre>

List the mount point

Action
<p>To verify the mount point was created, use this command syntax:</p> <pre>\$ server_mountpoint <movername> -list</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To verify the mount point was created, type:</p> <pre>\$ server_mountpoint server_2 -list</pre>
Output
<pre>server_2: /.etc_common /usr1 /usr1_ckpt1 /usr2 /usr3</pre>

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 33](#)
- ◆ [Mount a file system for NFSv4 access on page 34](#)
- ◆ [Unmount a file system on page 35](#)

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 (/).

Action
<p>To mount a file system on a Data Mover, use this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> /<mount_point></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = specifies mount options, separated by commas</p> <p><fs_name> = file system to be mounted</p> <p><mount_point> = path to mount point for the Data Mover</p> <p>Example:</p> <p>To mount a file system on server_2, type:</p> <pre>\$ server_mount server_2 ufs1 /ufs1</pre>
Output
<pre>server_2 : done</pre>

Verify the mount point

Action
<p>To verify whether the mount was created, use this command syntax:</p> <pre>\$ server_mount <movername></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To verify whether the mount was created, type:</p> <pre>\$ server_mount server_2</pre>
Output
<pre>server_2 : root_fs_2 on / uxfs,perm,rw root_fs_common on / .etc_common uxfs,perm,ro ufs1 on /ufs1 uxfs,perm,rw</pre>

Mount a file system for NFSv4 access

Use the `server_mount` command to mount a file system for NFSv4 access. [Planning considerations on page 19](#) 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 access control list (ACL) management
- ◆ Delegation mode

Action
<p>To mount a file system on a Data Mover, use this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> /<mount_point></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = specifies mount options, separated by commas</p> <p><fs_name> = file system to be mounted</p> <p><mount_point> = path to the mount point for the Data Mover</p> <p>Examples:</p>

Action
To mount a file system on server_2 and set the access policy to MIXED for file system ufs1, type: <pre>\$ server_mount server_2 -option accesspolicy=MIXED ufs1 /ufs1</pre> To mount a file system on server_2, disabling read-write delegation for file system ufs1, type: <pre>\$ server_mount server_2 -option nfsv4delegation=NONE ufs1 /ufs1</pre>
Output
<pre>server_2 : done</pre>

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.

Managing EMC Celerra Volumes and File Systems 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 a Celerra 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 19](#) 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 37](#)
- ◆ [Levels of access on page 38](#)
- ◆ [Multiple levels of a file system on page 39](#)
- ◆ [Restrict access by VLAN on page 40](#)
- ◆ [Specify NFSv4 access on page 41](#)
- ◆ [Specify the user authentication method on page 42](#)

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.

Action
<p>To export a Celerra file system for NFS access by client systems, use this command syntax:</p> <pre>\$ server_export <movername>-Protocol nfs [-name <name>] [-ignore] [-option <options>] [-comment <comment>] <pathname></pre> <p>where:</p> <p><code><movername></code> = name of the Data Mover</p> <p><code><name></code> = optional name for the NFS export</p> <p><code><options></code> = options to be applied to the NFS export</p> <p><code><comment></code> = optional comment</p> <p><code><pathname></code> = NFS export pathname</p> <p>Examples:</p> <p>To export a file system for NFS access, type:</p> <pre>\$ server_export server_2 -Protocol nfs -name nasdocsfs -comment 'NFS Export for ufs1' /ufs1</pre> <p>To export all file systems for NFS access, type:</p> <pre>\$ server_export server_2 -Protocol nfs -all</pre>
Output
<pre>server_2: done</pre>

Verify the file system exported

Action
<p>To verify whether the file system was exported, use this command syntax:</p> <pre>\$ server_export <movername> -list <pathname></pre> <p>where:</p> <p><code><movername></code> = name of the Data Mover</p> <p><code><pathname></code> = pathname of the NFS export. If no pathname is specified, all file system exports and shares are listed.</p> <p>Example:</p> <p>To verify whether the file system was exported, type:</p>

Action
<pre>\$ server_export server_2 -list /ufs1</pre>
Output
<pre>server_2 : export "/ufs1" name=/nasdocsfs comment="NFS Export for ufs1"</pre>

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 19](#) provides more information on understanding these options.

[Table 3 on page 38](#) describes the access options.

Table 3. NFS export access options

Options	Description
<code>ro</code>	Exports the path for all NFS clients as read-only.
<code>ro=<clients></code>	Exports the path for specified NFS clients as read-only.
<code>rw=<clients></code>	Exports the path as read or write for specified clients. If no other options are specified, all clients will have read-only access.
<code>access=<clients></code>	Provides default access for the specified clients. Denies access to those NFS clients who are not given explicit access.
<code>root=<clients></code>	Provides root access to clients listed in the export command, by specifying a client for <code>root=</code> . 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`.

[Appendix A](#) provides detailed information on how to set and interpret the access options.

Note: If you are planning to define an user authentication method by using the `sec` option, you must specify it before defining an access option. Otherwise, the export fails. [Appendix B](#) describes the rules to observe when defining the security options.

Specify levels of access

Action
<p>To specify an access level on an exported file system, use this command syntax:</p> <pre>\$ server_export movername -Protocol nfs [-option <options>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = export options, separated by commas</p> <p><pathname> = named path</p> <p>Examples:</p> <p>To export the file system /ufs1 as read-only for all NFS clients, type:</p> <pre>\$ server_export server_2 -Protocol nfs -option ro /ufs1</pre> <p>To export the file system /ufs1 as read-only for specified NFS clients, type:</p> <pre>\$ server_export server_2 -Protocol nfs -option ro, access=172.24.102.0/255.255.255.0 /ufs1</pre> <p>To export the file system /ufs1 as read-only and providing root privileges for a specified NFS client, type:</p> <pre>\$ server_export server_2 -Protocol nfs -option ro, root=172.24.102.240,access=172.24.102.0/255.255.255.0 /ufs1</pre>
Output
<pre>server_2 : done</pre>

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.

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

Action
<p>To export multiple levels of a Celerra file system for NFS access by client systems, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs [-name <name>] [-ignore] [-option <options>] [-comment <comment>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><name> = optional name for the NFS export</p> <p><options> = options to be applied to the NFS export</p> <p><comment> = optional comment</p> <p><pathname> = NFS export pathname</p> <p>Examples:</p> <p>To export a file system by giving read or write access to all clients, type:</p> <pre>\$ server_export server_2 -Protocol nfs /ufs1</pre> <p>To export a subdirectory of the previously exported path, by giving read only access to clients, type:</p> <pre>\$ server_export server_2 -Protocol nfs -option ro /ufs1/dir1</pre>
Output
<pre>server_2: done</pre>

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.

Action
<p>To export a Celerra file system for NFS access by client systems included in the specified VLANs, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs [-option <options>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = options to be applied to the NFS export</p> <p><pathname> = NFS export pathname</p> <p>Example:</p> <p>To export a file system for NFS access by client systems included in the specified VLANs, type:</p> <pre>\$ server_export server_2 -Protocol nfs -option vlan=102,103,104</pre>
Output
<pre>server_2: done</pre>

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. [Appendix B](#) describes the rules to observe when defining the security options.

Action
<p>To export a Celerra file system for NFS access, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs [-option <options>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = options to be applied to the NFS export</p> <p><pathname> = NFS export pathname</p> <p>Example:</p> <p>To export a file system for NFS access, type:</p>

Action
<code>\$ server_export server_2 -Protocol nfs -option nfsv4only /ufs1</code>
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 43](#) and ["Configure secure NFS using a UNIX or Linux Kerberos KDC" on page 54](#) provide more information on configuring secure NFS.

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.

Action
To export a Celerra file system with Kerberos authentication, use this command syntax:

Action
<pre>\$ server_export <movername> -Protocol nfs [-option <options>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = options to be applied to the NFS export</p> <p><pathname> = NFS export pathname</p> <p>Example:</p> <p>To export a file system with Kerberos authentication, type:</p> <pre>\$ server_export lngbe245 -Protocol nfs -option sec=krb5 /ufs1</pre>
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.

Action
<p>To export a Celerra file system with UNIX and Kerberos authentication, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs [-option <options>] <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = options to be applied to the NFS export</p> <p><pathname> = NFS export pathname</p> <p>Example:</p> <p>To export a file system by using UNIX and Kerberos authentication, type:</p> <pre>\$ server_export lngbe245 -Protocol nfs -option sec=krb5,sec=sys:ro /ufs1</pre>
Output
lngbe245: done

Configure secure NFS by using a Windows Kerberos KDC

Before you begin

Secure NFS is supported by using either of the following:

- ◆ Windows Kerberos KDC

- ◆ UNIX Kerberos KDC
- ◆ Linux Kerberos KDC

[Planning considerations on page 19](#) provides more information on these options.

Before configuring secure NFS in a Windows environment, you must configure access to a Windows 2000 or Windows Server 2003 domain, 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 EMC Celerra Time Services* 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 EMC Celerra Naming Services* describes how to configure access to NIS and iPlanet.

Procedure

To configure secure NFS using a Windows KDC:

- ◆ [Enable dynamic updates of reverse lookups on page 44](#)
- ◆ [Configure CIFS on page 46](#)
- ◆ [Set the secure NFS service instance on page 49](#)
- ◆ [Set the NFS user and group mapping service on page 52](#)
- ◆ [Reorder ACEs \(multiprotocol environment only\) on page 53](#)

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 *EMC Celerra Network Server Parameters Guide* 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.

Action

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
```

Action
where: <code><movername>=</code> name of the Data Mover
Example: To instruct the Data Mover's DNS client to update the PTR record for all CIFS servers, type: <code>\$ server_param server_2 -facility dns -modify updatePTRrecord -value 1</code>
Output
<code>server_2 : done</code>

Configure CIFS

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

Step	Action
1.	<p>Create a CIFS server on the Data Mover by using this command syntax:</p> <pre>\$ server_cifs <movername> -add compname=<comp_name>, domain=<full_domain_name></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><comp_name>= name of the server to be registered in DNS</p> <p><full_domain_name>= full domain name to which the server belongs. This means the name must contain at least one dot (.).</p> <p>Example:</p> <p>To create CIFS server cifsserver on server_2, type:</p> <pre>\$ server_cifs server_2 -add compname=cifsserver, domain=emc.com</pre> <p>Note: <i>EMC Celerra Network Server Command Reference Manual</i> describes the additional server_cifs command options. <i>Configuring and Managing CIFS on EMC Celerra</i> provides more information on CIFS configuration.</p>
2.	<p>Start the CIFS service to activate the CIFS protocol for the Data Mover by using this command syntax:</p> <pre>\$ server_setup <movername> -Protocol cifs -option start [=<n>]</pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><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)</p> <p>Example:</p> <p>To start the CIFS service on server_2, type:</p> <pre>\$ server_setup server_2 -Protocol cifs -option start</pre> <p>Note: The secure NFS service does not require additional CIFS threads.</p>

Step	Action
3.	<p>If not joined yet, join the CIFS server to a Windows domain by using this command syntax:</p> <pre>\$ server_cifs <movername> -Join compname=<comp_name>, domain=<full_domain_name>, admin=<domain_administrator_name>, ou=<organizational_unit></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><comp_name>= name of the server to be registered in DNS</p> <p><full_domain_name>= full domain name to which the server belongs. This means the name must contain at least one dot (.).</p> <p><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.</p> <p><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.</p> <p>Example:</p> <p>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:</p> <pre>\$ server_cifs server_2 -Join compname=cifsserver, domain=emc.com, admin=administrator,ou="ou=Computers: ou=Engineering"</pre>
4.	<p>Add the NFS service to the computer account by using this command syntax:</p> <pre>\$ server_cifs <movername> -Join compname=<comp_name>, domain=<full_domain_name>, admin=<domain_administrator_name>, ou=<organizational_unit> -options addservice=nfs</pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><comp_name>= name of the server to be registered in DNS</p> <p><full_domain_name>= full domain name to which the server belongs. This means the name must contain at least one dot (.).</p> <p><domain_administrator_name>= logon 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.</p> <p><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.</p> <p>Example:</p> <p>To add the NFS service to the server previously joined to the Windows domain, type:</p> <pre>\$ server_cifs server_2 -Join compname=cifsserver, domain=emc.com, admin=administrator -options addservice=nfs</pre>

Step	Action
5.	<p>Create a mount point by using this command syntax:</p> <pre>\$ server_mountpoint <movername> -create <pathname></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><pathname>= path to the mount point</p> <p>Example:</p> <p>To create a mount point called /usr1 on server_2, type:</p> <pre>\$ server_mountpoint server_2 -create /usr1</pre>
6.	<p>After you create a mount point, mount the file system by using this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> <mount_point></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><options>= specifies mount options, separated by commas</p> <p><fs_name>= file system to be mounted</p> <p><mount_point>= path to mount point for the Data Mover</p> <p>Example:</p> <p>To mount the file system ufs1 on server_2, type:</p> <pre>\$ server_mount server_2 -option accesspolicy=MIXED ufs1 /usr1</pre>
7.	<p>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:</p> <pre>\$ server_export <movername> -Protocol nfs <pathname></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><pathname>= path to the mount point</p> <p>Example:</p> <p>To create a file system for NFS access, type:</p> <pre>\$ server_export server_2 -Protocol nfs -name /usr1</pre>

Step	Action
8.	<p>(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:</p> <pre>\$ server_export <movername> -Protocol cifs -name <sharename> <pathname></pre> <p>where:</p> <p><movername>= name of the Data Mover</p> <p><sharename>= name of the CIFS share</p> <p><pathname> = path to the mount point</p> <p>Example:</p> <p>To create a share named cifs_share on server_2, type:</p> <pre>\$ server_export server_2 -Protocol cifs -name cifs_share /ufs1</pre>

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 the Celerra Network Server 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

Action
<p>To view the configuration of the secure NFS service:</p> <pre>\$ server_nfs <movername> -secnfs</pre> <p>where:</p> <p><movername> = name of the specified Data Mover</p> <p>Example:</p> <p>To view the configuration of the secure NFS service:s</p> <pre>\$ server_nfs server_2 -secnfsss</pre>

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

Step	Action
1.	<p>Delete the default secure NFS instance by using this command syntax:</p> <pre>\$ server_nfs <movename> -secnfs -principal -delete nfs@<server></pre> <p>where:</p> <p><movename> = name of the Data Mover</p> <p><server> = type of the realm</p> <p>Example:</p> <p>To delete the default secure NFS instance for server_2, type:</p> <pre>\$ server_nfs server_2 -secnfs -principal -delete nfs@server_2</pre> <p>Output:</p> <pre>server_2 : done</pre>

Step	Action
2.	<p>Add the new secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -principal -create nfs@<server></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><server> = type of the realm</p> <hr/> <p>Note: You must add the service twice using both formats: nfs@<Data_Mover_host_name> and nfs@<Data_Mover_fqdn></p> <hr/> <p>Example:</p> <p>To add a secure NFS service for the new hostname, type:</p> <pre>\$ server_nfs server_2 -secnfs -principal -create nfs@cifsserver</pre> <p>Output:</p> <pre>server_2 : done</pre> <p>Add a second service instance by using the server's fully-qualified domain name, type:</p> <pre>\$ server_nfs server_2 -secnfs -principal -create nfs@cifsserver.emc.com</pre>
3.	<p>Stop the secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -service -stop</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To stop the secure NFS service for server_2, type:</p> <pre>\$ server_nfs server_2 -secnfs -service -stop</pre> <p>Output:</p> <pre>server_2 : done</pre>
4.	<p>Start the secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -service -start</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To start the secure NFS service for server_2, type:</p> <pre>\$ server_nfs server_2 -secnfs -service -start</pre> <p>Output:</p> <pre>server_2 : done</pre>

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 the Celerra Network Server 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 using the Kerberos security option, the system authenticates the principal. 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 Celerra 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.

Action
<p>To verify the type of mapping service used by secure NFS, use this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -mapper -info</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p>

Action
To list the mapping service for server_2, type: \$ server_nfs server_2 -secnfs -mapper -info
Output
<pre>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</pre>

Set automatic mapping

Action
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

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

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 the ACE order is not the one it prefers, it

displays a pop-up on the Windows client warning 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 *EMC Celerra Network Server Parameters Guide* provides additional information about the `acl.sortAces` parameter.

Sort ACEs

Note: Parameter and facility names are case-sensitive.

Action
<p>To sort ACEs in the order expected by Windows Explorer, use this command syntax:</p> <pre>\$ server_param <movername> -facility cifs -modify acl.sortAces -value 1</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To set the cifs <code>acl.sortAces</code> parameter to 1, type:</p> <pre>\$ server_param server_2 -facility cifs -modify acl.sortAces -value 1</pre>
Output
<pre>server_2 : done</pre>

Configure secure NFS 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 19](#) 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 EMC Celerra Time Services* describes how to configure time services.
- ◆ For the resolution of users to UIDs and groups to GIDs, EMC recommends 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 EMC Celerra Naming Services* describes how to configure access to NIS and iPlanet.

Procedure

To configure secure NFS:

- ◆ [Set the Data Mover name on page 55](#)
- ◆ [Configure the Kerberos realm on page 56](#)
- ◆ [Set the secure NFS service instance on page 57](#)
- ◆ [Create the NFS Kerberos service principals on page 61](#)
- ◆ [Map user principal names to UIDs on page 64](#)

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 Celerra 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.

Action
<p>To set the Data Mover's name, use this command syntax:</p> <pre>\$ server_name <movername> <new_name></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><new_name> = new name for the Data Mover</p> <p>Example:</p> <p>To rename the Data Mover, server_2 to lngbe245, type:</p> <pre>\$ server_name server_2 lngbe245</pre>

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.

Action

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>`= fully-qualified domain name (FQDN) of the Kerberos realm to be added to the key distribution center (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, defaultrealm
```

Output

```
lngbe245 : done
```

Verify the results

Action

To verify the results, use this command syntax:

```
$ server_kerberos movername -list
```

Action
<p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To list the domain and realm information for server_2, type:</p> <pre>\$ server_kerberos lngbe24 -list</pre>
Output
<pre>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 kadmind: 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</pre>

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 Celerra Network Server 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

Action
To view the configuration of the secure NFS service:

Action
<pre>\$ server_nfs <movername> -secnfs</pre> <p>where:</p> <p><movername> = name of the specified Data Mover</p> <p>Example:</p> <p>To view the configuration of the secure NFS service:s</p> <pre>\$ server_nfs server_2 -secnfsss</pre>
Output
<pre>server_2 : RPCSEC_GSS server stats Credential count: 1 principal: nfs@server_2 Total number of user contexts: 0 Current context handle: 1</pre>

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.

Step	Action
1.	<p>Delete the default secure NFS instance by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -principal -delete <service@server></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><service@server> = the type of service and the realm</p> <p>Example:</p> <p>To delete the secure NFS instance for server_2, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -principal -delete nfs@server_2</pre> <p>Output:</p> <pre>lngbe245 : done</pre>

Step	Action
2.	<p>Add the new secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -principal -create nfs@<server></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><server> = type of the realm</p> <hr/> <p>Note: You must add the service twice by using both formats: nfs@<Data_Mover_host_name> and nfs@<Data_Mover_fqdn></p> <hr/> <p>Example:</p> <p>To add a secure NFS service for the new hostname, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -principal -create nfs@lngbe245</pre> <p>To add a second service instance that uses the server's fully-qualified domain name, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -principal -createlnge245.lss.exam.com</pre> <p>Output:</p> <pre>lngbe245 : done</pre>
3.	<p>Stop the secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -service -stop</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To stop the secure NFS service for lngbe245, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -service -stop</pre> <p>Output:</p> <pre>lngbe245 : done</pre>
4.	<p>Start the secure NFS service by using this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -service -start</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To start the secure NFS service for lngbe245, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -service -start</pre> <p>Output:</p> <pre>lngbe245 : done</pre>

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.

Action
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 61 .
Use the server_file command and FTP to copy the file (binary transfer) to the /etc directory on the Kerberos KDC.

Note

The *EMC Celerra Network Server Command Reference Manual* provides more information about using the `server_file` command. *Using FTP on EMC Celerra Network Server* provides information on using FTP.

Service principals

Action

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>`), type:

```
kadmin: add princ -randkey nfs/lngbe245
```

To create the other service principal, the FQDN for the Data Mover, `nfs/<Data_Mover_fqdn>`, type:

```
kadmin: addprinc -randkey nfs/lngbe245.1ss.exam.com
```

Output

```
Principal "nfs/lngbe245.1ss.exam.com@example.com" created.
```

Verify the principals

Action

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_1.1ss.exam.com@example.com
root/kdc_1.1ss.exam.com@example.com changepw/
mzolla@example.com
kadmin/changepw@example.com
kadmin/changepw@example.com

nfs/lngbe245@example.com

nfs/lngbe245.1ss.exam.com@example.com
```

Note: The realm is automatically added to the service principals.

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.

Action
<p>To generate keys for each service principal and add them to the keytab file, use this command syntax:</p> <pre>kadmin: ktadd -k <keytab_file_path> nfs/ <name></pre> <p>where:</p> <p><keytab_file_path> = the location of the keytab file</p> <p><name> = the name of the previously created service principal</p> <p>Example:</p> <p>To generate keys for <code>lngbe245</code> and add them to <code>tmp/krb5.keytab</code>, type:</p> <pre>kadmin: ktadd -k /tmp/krb5.keytab nfs/lngbe245</pre>
Output
<pre>Entry for principal nfs/lngbe245..lss.exam.com with kvno 5, encryption type DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.</pre>

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

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 Celerra Network Server Command Reference Manual* 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.

Action
<p>To view the keytab information, type:</p> <pre>\$ server_kerberos lngbe245 -keytab</pre>

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 Celerra file system is based on UIDs and GIDs. The user's Kerberos principal name is mapped to the UID using a mapping or naming service. During user authentication, this mapping information is used to determine data access to Kerberos-enabled services.

[Table 4 on page 64](#) 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 EMC Celerra 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 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 Celerra Network Server 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 only seen during the initialization of secure NFS mapping.*

Action
<p>To verify the type of mapping service used by secure NFS, use this command syntax:</p> <pre>\$ server_nfs <movename> -secnfs -mapper -info</pre> <p>where:</p> <p><code><movename></code> = name of the Data Mover</p> <p>Example:</p> <p>To list the mapping service for <code>lngbe245</code>, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -mapper -info</pre>

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

Action

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
```

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

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 running gsscred, can be stored on a NIS server (recommended) or stored locally on the Data Mover.

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

Table 5 on page 67 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.

Step	Action
1.	To specify the mapping configuration as NIS, type: <pre>\$ server_nfs lngbe245 -secnfs -mapper -set -source nis</pre> Output: lngbe245: done
2.	To set the location of the password database, type: <pre>\$ server_nfs lngbe245 -secnfs -mapper -set -passwddb nis</pre> Output: lngbe245: done

Step	Action
3.	<p>To verify the mapping configuration for NIS, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -mapper -info</pre> <p>Output:</p> <pre>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</pre>
4.	From the NIS master server, build the NIS map for gsscred_db. NIS documentation has more information.

Note: *Configuring EMC Celerra Naming Services* provides additional information about NIS and the Celerra Network Server.

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 on EMC Celerra Network Server describes how to use FTP.

Action
<p>To copy the mapping file to the Data Mover, using this command syntax:</p> <pre>\$ server_file <movername> -put <src_file> <dst_file></pre> <p>where:</p> <pre><movername> = name of the Data Mover <src_file> = source file <dst_file> = destination file</pre> <p>Example:</p> <p>To copy the mapping file gsscred_db to the Data Mover, type:</p> <pre>\$ server_file lngbe245 -put gsscred_db gsscred_db</pre>

Output

```
lmgbe245 : done
```

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

Create a local mapping file

Before you begin

Use the `server_nfs <movename> -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.

Procedure

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

Action

To create a mapping entry, use this command syntax:

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

where:

`<user_name>` = the username

`<UID>` = user ID

Example:

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

```
$ server_nfs lmgbe245 -secnfs -mapper -mapping -create name=nfsuser1
```

Output

```
lmgbe245 : done
```

Note

Because the format of the gsscred_db map differs for a Solaris and a Celerra, 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.

Action

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:

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

Output

```
lngbe245 : done
```

Verification

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
```

Verify the mapping configuration

Action
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 Celerra file system on a client system

To allow NFS users to access a Celerra file system, the exported Celerra 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.

The tasks to manage NFS are:

- ◆ [Manage NFS on page 74](#)
- ◆ [Manage NFSv4 on page 82](#)
- ◆ [Manage Secure NFS on page 89](#)

Manage NFS

Tasks to manage basic NFS access to file systems are:

- ◆ [Unexport the NFS path to a Celerra file system on page 74](#)
- ◆ [Reexport all NFS paths on a Celerra Network Server on page 75](#)
- ◆ [Disable NFS access to all file systems on a Data Mover on page 75](#)
- ◆ [NFS automounter feature support on page 76](#)

Unexport the NFS path to a Celerra file system

Unexport the NFS path on the Data Mover to stop NFS client access to a Celerra 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

Action
<p>To temporarily unexport an NFS path to a Celerra file system, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -unexport <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><pathname> = the NFS export pathname</p> <p>Example:</p> <p>To temporarily unexport the /ufs1 NFS path, type:</p> <pre>\$ server_export server_2 -Protocol nfs -unexport /ufs1</pre>
Output
<pre>server_2: done</pre>

Permanently unexport an NFS path

Action
<p>To permanently unexport an NFS path to a Celerra file system, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -unexport -perm <pathname></pre>

Action
<p>where:</p> <p><movername> = name of the Data Mover</p> <p><pathname> = the NFS export pathname</p> <p>Example:</p> <p>To permanently unexport the /ufs1 NFS path, type:</p> <pre>\$ server_export server_2 -Protocol nfs -unexport -perm /ufs1</pre>
Output
<pre>server_2: done</pre>

Reexport all NFS paths on a Celerra Network Server

You can reexport all NFS paths to the Celerra file systems on a Celerra Network Server 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.

Action
<p>To reexport all NFS paths from a Celerra Network Server, type:</p> <pre>\$ server_export ALL -Protocol nfs -all</pre>
Output
<pre>server_2 : done server_3 : done server_4 : done</pre>

Disable NFS access to all file systems on a Data Mover

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

You can stop NFS access to the Celerra 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

Action
<p>To temporarily unexport all NFS paths on a Data Mover, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -unexport -all</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To temporarily unexport all NFS paths on server_2, type:</p> <pre>\$ server_export server_2 -Protocol nfs -unexport -all</pre>
Output
<pre>server_2: done</pre>

Permanently unexport all NFS paths

Action
<p>To permanently unexport all NFS paths on a Data Mover, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -unexport -perm -all</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To permanently unexport all NFS paths on server_2, type:</p> <pre>\$ server_export server_2 -Protocol nfs -unexport -perm -all</pre>
Output
<pre>server_2: done</pre>

NFS automounter feature support

If you use the NFS automounter feature on NFS clients, you can generate an automount map file on the Celerra Network Server 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 Celerra Network Server, 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

Action
<p>To create an automount map file and print it to the screen, type:</p> <pre>\$ nas_automountmap -create</pre> <p>To create and save an automount map file, use this command syntax:</p> <pre>\$ nas_automountmap -create -out <outfile></pre> <p>where:</p> <p><outfile> = name of the output file</p> <p>Example:</p> <p>To create and save an automount map file named automountmap, type:</p> <pre>\$ nas_automountmap -create -out automountmap</pre>

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

Verification
<p>To view the automount map file and confirm it was saved, use this command syntax:</p> <pre>\$ more <outfile></pre> <p>where:</p> <p><outfile> = name of the output file</p> <p>Example:</p> <p>To view the automount map file and confirm it was saved as automountmap, type:</p> <pre>\$ more automountmap</pre>

Output
<pre>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</pre>

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.

Action
<p>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:</p> <pre>\$ nas_automountmap -create -in <infile> -out <outfile></pre> <p>where:</p> <p><infile> = name of the automount map file</p> <p><outfile> = name of the new output file</p> <p>For example:</p> <p>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:</p> <pre>\$ nas_automountmap -create -in automountmap -out automountmap1</pre>
Output
<p>Create and save merged automount map file output. No output to screen.</p>

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.

View the updated automount map file

Verification
<p>To view the updated automount map file, use this command syntax:</p> <pre>\$ more <outfile></pre> <p>where:</p> <p><outfile> = name of the output file</p> <p>Example:</p> <pre>\$ more automountmap1</pre>

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.

Action

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

Action

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
<pre>Conflicting lists: ufs1 -rw,intr,suid 172.16.21.202:/ufs1 ufs1_172.16.21.203 -rw,intr,suid 172.16.21.203:/ufs1</pre>

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 param can be used to filter out entries if the client does not have mount permissions for the filesystem 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 param 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 param forceFullShowmount takes two values: 0 or 1, with the default being 1 (show all).

Action
<p>To hide NFS exports, use this command syntax:</p> <pre>\$ server_param <movername> -facility mount -modify forceFullShowmount -value 0</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To hide NFS exports, type:</p> <pre>\$ server_param server_2 -facility mount -modify forceFullShowmount -value 0</pre>
Output
<pre>server_2 : done</pre>

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.

Action
<p>To verify whether the forceFullShowmount was set, use this command syntax:</p> <pre>\$ server_param <movername> -facility mount -info forceFullShowmount -verbose</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To verify whether the forceFullShowmount was set , type:</p> <pre>\$ server_param server_2 -facility mount -info forceFullShowmount -verbose</pre>
Output
<pre>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.</pre>

Manage NFSv4

Tasks to manage NFSv4 access to file systems are:

- ◆ [Modify state duration on page 82](#)
- ◆ [Change the delegation recall timeout on page 83](#)
- ◆ [Modify number of usable nodes on page 83](#)
- ◆ [Display NFSv4 service status on page 84](#)
- ◆ [Stop the NFSv4 service on page 84](#)
- ◆ [Restart the NFSv4 service on page 85](#)
- ◆ [Display NFSv4 clients on page 85](#)
- ◆ [Display information about NFSv4 clients on page 86](#)
- ◆ [Release NFSv4 clients on page 88](#)
- ◆ [Support 32-bit and 64-bit NFs clients on page 88](#)

Modify state duration

The *EMC Celerra Network Server Parameters Guide* provides additional information about the `leaseDuration` parameter and its values. Parameter and facility names are case-sensitive.

Action
<p>To modify the time during which the server maintains client states, use this command syntax:</p> <pre>\$ server_param <movename> -facility nfsv4 -modify leaseDuration -value <new_value></pre> <p>where:</p> <p><movename> = name of the Data Mover</p> <p><new_value> = This value must be less than the grace period duration specified by the lockd facility <code>gpDuration</code> parameter.</p> <p>Example:</p> <p>To modify the time during which the server maintains client states, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -modify leaseDuration -value 20</pre>
Output
<pre>server_2 : done</pre>

Change the delegation recall timeout

The *EMC Celerra Network Server Parameters Guide* provides additional information about the `recallTimeout` parameter. Parameter and facility names are case-sensitive.

Action
<p>To modify the time the server waits before recalling delegations, use this command syntax:</p> <pre>\$ server_param <movername> -facility nfsv4 -modify recallTimeout -value <new_value></pre> <p>where:</p> <p><i>movername</i> = name of the Data Mover</p> <p><i>new_value</i> = value for the parameter in seconds. The range value is 5-60.</p> <p>Example:</p> <p>To modify the time the server waits before recalling delegations, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -modify recallTimeout -value 20</pre>
Output
<pre>server_2 : done</pre>

Modify number of usable nodes

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

Action
<p>To modify the maximum percentage of nodes which the NFSv4 server utilizes, use this command syntax:</p> <pre>\$ server_param <movername> -facility nfsv4 -modify vnodePercent -value <new_value></pre> <p>where:</p> <p><i><movername></i> = name of the Data Mover</p> <p><i><new_value></i> = value for the parameter in seconds. 10-80 is the range of value</p> <p>Example:</p> <p>To modify the maximum percentage of nodes which the NFSv4 server utilizes to 50%, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -modify vnodePercent -value 50</pre>
Output
<pre>server_2 : done</pre>

Note: Restart the Data Mover for the change to take effect.

Display NFSv4 service status

Action
<p>To display NFSv4 service status, use this command syntax:</p> <pre>\$ server_nfs <movername> -v4</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To display NFSv4 service status on server_2, type:</p> <pre>\$ server_nfs server_2 -v4</pre>
Output
<pre>server_2 : -----nfsv4 server status ----- * Service Started * ----- NFSv4 Clients ----- Confirmed Clients : 2 UnConfirmed Clients : 0 ----- ----- NFSv4 State ----- Opens : 4 Locks : 0 Delegations : 0 -----</pre>

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 Celerra 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.

Action
<p>To stop the NFSv4 service, use this command syntax:</p> <pre>\$ server_nfs <movername> -v4 -service -stop</pre> <p>where:</p>

Action
<p><code><movername></code> = name of the Data Mover</p> <p>Example:</p> <p>To stop the NFSv4 service on server_2, type:</p> <pre>\$ server_nfs server_2 -v4 -service -stop</pre>
Output
<pre>server_2 : done</pre>

Restart the NFSv4 service

The NFSv4 service must be enabled before you can start it. [Getting started on page 28](#) describes how to enable NFSv4 support. This command stops the NFSv4 service only. Other versions of NFS continue to run.

Action
<p>To restart the NFSv4 service, enable it and use this command syntax:</p> <pre>\$ server_nfs <movername> -v4 -service -start</pre> <p>where:</p> <p><code><movername></code> = name of the Data Mover</p> <p>Example:</p> <p>To start the NFSv4 service on server_2, type:</p> <pre>\$ server_nfs server_2 -v4 -service -start</pre>
Output
<pre>server_2 : done</pre>

Display NFSv4 clients

Action
<p>To display all client systems that have an established state with the NFSv4 server, type:</p> <pre>\$ server_nfs <movername> -v4 -client -list</pre> <p>where:</p> <p><code><movername></code> = name of the Data Mover</p> <p>Example:</p> <p>To display all client systems that have an established state with the NFS service on server_2, type:</p>

Action
<pre>\$ server_nfs server_2 -v4 -client -list</pre>
Output
<pre>server_2 : ----- nfsv4 server client list ----- Hostname/ip: Index win901234 : 0xe2400000 10.171.2.76 : 0xef400000</pre>

Note
<p>The first column displays the NFSv4 client hostname or IP address. The second column displays an index number that Celerra uses to identify the client connection.</p> <p>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.</p> <p>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.</p>

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.

Action
<p>To display information about a client system that has an established state with the NFSv4 server, type:</p> <pre>\$ server_nfs <movername> -v4 -client -info { index=<index> hostname=<host> IPaddress=<addr> }</pre> <p>where:</p> <p><code><movername></code> = name of the Data Mover</p> <p><code><index></code> = index number assigned to the client system</p> <p><code><host></code> = hostname of the client system</p> <p><code><addr></code> = IP address of the client system</p> <p>Examples:</p> <p>To display information about a client system identified by its index number that has an established state with the NFS service on <code>server_2</code>, type:</p> <pre>\$ server_nfs server_2 -v4 -client -info index=0xe2400000 \$ server_nfs server_2 -v4 -client -info index=0xef400000</pre>

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 Celerra 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.

Action
<p>To release the client ID of a client system that has an established state with the NFSv4 server, use this command syntax:</p> <pre>\$ server_nfs <movername> -v4 -client -release { index=<index> hostname=<host> IPaddress=<addr> }</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><index> = index number assigned to the client system</p> <p><host> = hostname of the client system</p> <p><addr> = IP address of the client system</p> <p>Example:</p> <p>To release a client system identified by its index number that has an established state with the NFS service on server_2, type:</p> <pre>\$ server_nfs server_2 -v4 -client -release index=0xe2400000</pre>
Output
<pre>server_2 : done</pre>

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 Celerra NFSv4 server defaults to 32 bits.

The *EMC Celerra Network Server Parameters Guide* provides additional information about the 32bitClient parameter .

Note: Parameter and facility names are case-sensitive.

Action
<p>To enable the NFSv4 server to support 64 bits, use this command syntax:</p> <pre>\$ server_param <movename> -facility nfsv4 -modify 32bitClient -value 0</pre> <p>where:</p> <p><movename> = name of the Data Mover</p> <p>Example:</p> <p>To enable the NFSv4 server to support 64 bits, type:</p> <pre>\$ server_param server_2 -facility nfsv4 -modify 32bitClient -value 0</pre>
Output
<pre>server_2 : done</pre>

Manage Secure NFS

Tasks to manage secure NFS are:

- ♦ [View keytab entries on page 89](#)
- ♦ [Display all secure NFS service instances on page 90](#)
- ♦ [Display user attributes on page 91](#)
- ♦ [Display information about a user in a local mapping file on page 92](#)
- ♦ [Delete a user from a local mapping file on page 93](#)
- ♦ [Delete service principals on page 93](#)
- ♦ [Release authentication on page 94](#)

View keytab entries

Action
<p>To view the keytab information, type:</p> <pre>\$ server_kerberos lngbe245 -keytab</pre>

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

Action

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

```
$ server_nfs <movename> -secnfs -user -list
```

where:

<movename> = name of the Data Mover

Example:

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

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

Output
<pre> 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 </pre>

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.

Action
<p>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:</p> <pre> \$ server_nfs <movername> -secnfs -user -info { principal=<user_principal> handle=<handle_number> } </pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><user_principal> = name of the user</p> <p><handle_number> = the user's ID</p> <p>Example:</p> <p>To display the attributes of an authenticated user, type:</p> <pre> \$ server_nfs lngbe245 -secnfs -user -info handle=38 </pre>

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 64](#) provides more information about using a local mapping file.

Action

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

```

$ server_nfs <mover_name> -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
 737573657231407374617465732E656D632E636F6D
 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 64](#) provides more information about using a local mapping file.

Action
<p>To delete a user entry from a local mapping file, use this command syntax:</p> <pre>\$ server_nfs -secnfs -mapper -mapping -delete { name=<user_name> uid=<UID>}</pre> <p>where:</p> <p><user_name> = username</p> <p><UID> = user ID</p> <p>Example:</p> <p>To delete a user nfsuser1 from the local mapping file, type:</p> <pre>\$ server_nfs -secnfs -mapper -mapping -delete name=nfsuser1</pre>
Output
<pre>lngbe245: done</pre>

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.

Action
<p>To delete a service principal, use this command syntax:</p> <pre>\$ server_nfs <movename> -secnfs -principal -delete nas@<server></pre> <p>where:</p> <p><movename> = name of the Data Mover</p> <p><server> = type of the realm</p> <p>Example:</p> <p>To delete a service principal, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -principal -delete nfs@lngbe245</pre>
Output
<pre>lngbe245: done</pre>

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 using the `-list` option.

Action
<p>To release the secure NFS authentication context for a user, use this command syntax:</p> <pre>\$ server_nfs <movername> -secnfs -user -release { principal=<user_principal> handle=<handle_number>}</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><user_principal> = name of the released principal</p> <p><handle_number> = the user ID</p> <p>Example:</p> <p>To release authentication for the <code>nfsuser1</code>, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -user -release principal=nfsuser1@secnfs.eng</pre>
Output
<pre>lngbe245: done</pre>

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 representative.

- ◆ [EMC E-Lab Interoperability Navigator on page 96](#)
- ◆ [Troubleshooting NFSv4 on page 96](#)
- ◆ [Troubleshooting secure NFS on page 102](#)
- ◆ [Error messages on page 105](#)
- ◆ [NFSv4 error messages on page 105](#)
- ◆ [EMC Training and Professional Services on page 105](#)

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 at <http://Powerlink.EMC.com>. After logging in to Powerlink, go to **Support** ► **Interoperability and Product LifeCycle Information** ► **E-Lab Interoperability Navigator**.

Troubleshooting NFSv4

When you encounter problems using NFSv4, review the following:

- ◆ The netd 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

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

Action
<pre>\$ server_nfs <movername> -v4</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To display the status of the NFSv4 service on server_2, type:</p> <pre>\$ server_nfs server_2 -v4</pre>
Output
<pre>server_2: ----- nfsv4 server status ----- * service started * ----- nfsv4 clients ----- configured clients: 5 unconfirmed clients: 0 ----- ----- nfsv4 state ----- opens: 8 locks: 4 delegations: 0</pre>

Display NFS statistics

Action
<p>To display NFS statistics, use this command syntax:</p> <pre>\$ server_nfs <movername> -stats</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To display NFS statistics, type:</p> <pre>\$ server_nfs server_2 -stats</pre>

```

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
    
```

Output						
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

Output						
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)						

Output

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 using secure NFS, review the:

- ◆ 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.

Action
To retrieve a list of users and principals, type: server_nfs lngbe245 -secnfs -user -list
Output
<pre>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</pre>

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.

Action
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

Action
<p><user_principal> = the released principal</p> <p>Example:</p> <p>To release authentication for nsfuser1, type:</p> <pre>\$ server_nfs lngbe245 -secnfs -user -release principal=nsfuser1@secnfs.eng</pre>
Output
lngbe245: done

Error messages

As of version 5.6, all new 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:

- ◆ Celerra Manager:
 - Right-click an event, alert, or status message and select to view Event Details, Alert Details, or Status Details.
- ◆ Celerra CLI:
 - Type `nas_message -info <MessageID>`, where `<MessageID>` is the message identification number.
- ◆ *EMC Celerra Network Server Error Messages Guide*:
 - Use this guide to locate information about messages that are in the earlier-release message format.
- ◆ Powerlink:
 - Use the text from the error message's brief description or the message's ID to search the Knowledgebase on [Powerlink](#). After logging in to Powerlink, go to **Support** ► **Knowledgebase Search** ► **Support Solutions Search**.

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 in order 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 Powerlink at <http://Powerlink.EMC.com> for course and registration information.

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

Appendix A

System Access Behavior

This appendix describes how to interpret the read and write access 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.

- ◆ [Behavior when specifying combinations of access modes on page 108](#)
- ◆ [Rules for resolving conflicts among hosts, subnets, and netgroups on page 109](#)
- ◆ [Specify read-only as the default access mode on page 111](#)

Behavior when specifying combinations of access modes

The table shows what happens when specifying combinations of access modes for the `server_export` command. An X in the table column means 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. The following table 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
Hosts and subnets have preference over netgroups	<pre>rw=host1, ro=netgrp1 rw=netgrp1, ro=subnet1</pre>	<p>Host1 is granted read/write access.</p> <p>All other hosts belonging to <code>netgrp1</code> are granted read-only access.</p> <p>Host1 is granted read-only access.</p> <p>All hosts belonging to <code>subnet1</code> including <code>host1</code> are granted read-only access.</p> <p>All hosts belonging to <code>netgrp1</code> except <code>host1</code> are granted read or write access.</p>
<p>If you use read or write and read-only access, read/write access is granted by default.</p> <p>To specify read-only as the default access, specify the <code>secureExportMode</code> parameter.</p> <p>"Specify read-only as the default access mode" on page 111 provides information about setting this parameter.</p>	<pre>rw=host1,ro=host1 ro=subnet1,rw=subnet1</pre>	<p>Host1 is granted read or write access.</p> <p>All hosts belonging to <code>subnet1</code> are granted read/write access.</p> <p>_____</p> <p>Note: If the parameter is set, <code>host1</code> is granted read-only access.</p> <p>_____</p>
If you use a host and a subnet, the first matching entry is used for granting access.	<pre>rw=-subnet1:host1 rw=host1:-subnet1</pre>	<p>Host1 is denied read/write access.</p> <p>All other hosts belonging to <code>subnet1</code>, including <code>Host1</code> are denied read/write access.</p> <p>Host1 is granted read/write access.</p> <p>All other hosts belonging to <code>subnet1</code> are denied read/write access.</p> <p>_____</p> <p>Note: Negation is used in this example.</p> <p>_____</p>

Table 7. Rules for resolving conflicts (continued)

Rule	Example	Access
If there are two hosts, the first matching entry is used for granting access.	rw=-host1:host1 rw=host1:-host1	Host1 is denied read/write access. 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 rw=subnet2:-subnet1	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. 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 rw=netgrp2:-netgrp1	Host1 is denied read/write access. All other hosts belonging to netgrp1 are denied read/write access. All other hosts belonging to netgrp2 are granted read/write access. 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.

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 enabling read-only access when a conflict exists by changing the `secureExportMode` parameter. [Rules for resolving conflicts among hosts, subnets, and netgroups on page 109](#) provides more information.

The *EMC Celerra Network Server Parameters Guide* provides additional information about the `secureExportMode` parameter. Parameter and facility names are case-sensitive.

Action
<p>To turn on secure export mode, use this command syntax:</p> <pre>\$ server_param <movename> -facility nfs -modify secureExportMode -value 1</pre> <p>where:</p> <p><movename> = name of the Data Mover</p> <p>Example:</p> <p>To set the NFS <code>secureExportMode</code> parameter to 128, type:</p> <pre>\$ server_param server_2 -facility nfs -modify secureExportMode -value 1</pre>
Output
<pre>server_2 : done</pre>

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.

- ◆ [General rules when specifying security options on page 114](#)
- ◆ [Root access mode on page 115](#)

General rules when specifying security options

The following rules apply when specifying the security option. [Levels of access on page 38](#) provides information about access mode interaction when exporting.

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 using the AUTH_SYS method.	<pre>server_export lngbe245 -Protocol nfs -option ro /ufs</pre>
If you specify an access mode before the security option, the export fails.	<pre>server_export lngbe245 -Protocol nfs -option rw=client1:client2, sec=krb5:ro=client3 /ufs1</pre>
If you repeat the value of a security option, the export fails.	<pre>server_export lngbe245 -Protocol nfs -option sec=krb5 :rw=client1, sec=krb5 :ro=client2 /ufs1</pre>
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.	<pre>server_export lngbe245 -Protocol nfs -option sec=krb5,sec=sys :ro /ufs1</pre>
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.	<pre>server_export lngbe245 -Protocol nfs -option sec=krb5: rw=client1:client2, ro=client3:client4 /ufs1</pre>

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 whether access is 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 the following table.

Table 9. Root access and secure NFS

Command example	Explanation
<pre>server_export lngbe245 -Protocol nfs -option sec=krb5:root=client1,sec=sys:ro /ufs1</pre>	<p>Root user on client1 can access the file system.</p> <p>If the root user authenticates using Kerberos security, read/write access is granted.</p> <p>If the user authenticates using the default security, read-only access is granted.</p>
<pre>server_export lngbe245 -Protocol nfs -option sec=krb5:root=client1, sec=sys:access=client1 /ufs1</pre>	<p>All clients authenticated using Kerberos are granted read/write access.</p> <p>Root user from client1 is also granted access with root privileges.</p>
<pre>server_export lngbe245 -Protocol nfs -option sec= krb5:ac cess=client1,root=client2 /ufs1</pre>	<p>Only client1 has access to the file system.</p> <p>All other clients, including client2, are denied access. For client2 to have access, it needs to be on the access list.</p>

NFS Authentication Daemon for PC Clients

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

- ◆ [PC client access on page 118](#)
- ◆ [Set up PC client software on page 119](#)
- ◆ [Hummingbird PC NFS client issues on page 119](#)

PC client access

Similar to native UNIX and Linux clients, a PC client that uses NFS to access Celerra file systems must be authenticated successfully before it can get access to the Celerra Network Server, as shown in [Figure 2 on page 118](#).

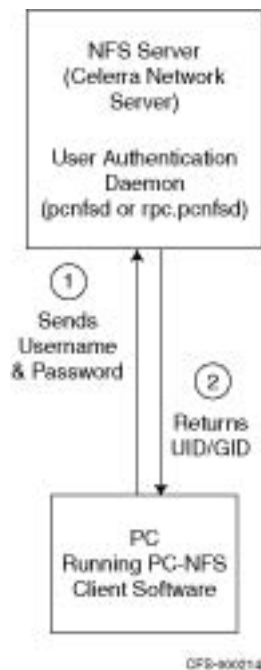


Figure 2. PC client access

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 Celerra Network Server.

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

If your PC software configuration does not include NFS software and you want to use the PC to access Celerra file systems using 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 Celerra Network Server. 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.

Step	Action
1.	Set up a user account. <code>server_user</code> command has further information.
2.	Open the <code>/nas/server/server_<x>/netd</code> file with a text editor, add <code>pcnfs</code> on a separate line after the <code>nfs</code> start line, and save the file. where <code><x></code> is the 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 36 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 running <code>pcnfsd</code> , which returns the UID or GID numbers for the PC client.

Note: 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 running `pcnfsd` in the subnet, while NFS Maestro has an option where you specify the system's name.

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 Celerra (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.

A

access control list (ACL)

List of access control entries (ACEs) that provide information about the users and groups allowed access to an object.

authentication

Process for verifying the identity of a user trying to access a resource, object or service, such as a file or a directory.

C

Celerra Data Migration Service (CDMS)

Feature for migrating file systems from NFS and CIFS source file servers to a Celerra Network Server. The online migration is transparent to users once it starts.

Celerra FileMover

Policy-based system used to determine where files should be physically stored. In most cases, policies are based on file size or last access time (LAT) or both and are used to identify data that can be moved to slower, less-expensive storage.

CIFS

See Common Internet File System.

D

DNS

See Domain Name System.

F

fully qualified domain name (FQDN)

Full name of a system, which contains the domain name of the organization and its highest subdomain.

G

GID

See group identifier.

K

Kerberos

Authentication, data integrity, and data privacy encryption mechanism used to encode authentication information. Kerberos coexists with NTLM (Netlogon services) and, using secret-key cryptography, provides authentication for client/server applications.

N

Network Information Service (NIS)

Distributed data lookup service that shares user and system information across a network, including usernames, passwords, home directories, groups, hostnames, IP addresses, and netgroup definitions.

R

realm

In a networked environment, a set of security principals (users, service, computers) subject to Kerberos authentication and managed by the same Kerberos authority.

S

storage system

Array of physical disk devices and their supporting processors, power supplies, and cables.

U

UID

See User ID.

Unicode

Family of universal character encoding standards used for representation of text for computer processing.

Unicode or UCS Transformation Format-8 (UTF-8)

Multibyte encoding form, UTF-8 uses an algorithmic mapping scheme to convert every Unicode value to a unique 1- to 4-byte sequence with no embedded null characters.

32bitClient parameter 88

A

access control lists (ACLs) 21
access, disabling NFS 75
acl.sortAces parameter 54
authentication
 for PC clients 19
 for UNIX and Linux clients 19
automounter feature 76

C

CDMS 13
Celerra FileMover feature 13
 parameters
 acl.sortAces 53
client contexts 25

D

delegation 23
disabling NFS access 75
DNS
 configuring for secure NFS 54
 parameters
 updatePTRrecord 44

E

exporting
 using Kerberos authentication 42
 using UNIX and Kerberos authentication 43
 using UNIX authentication 42

F

file locking 24
file systems
 exporting with Kerberos authentication 42
 exporting with UNIX and Kerberos authentication 43
 exporting with UNIX authentication 42
 mounting 33

H

hivers option 28

I

internationalization 22

K

Kerberos
 using a UNIX or Linux KDC with secure NFS 54
 using a Windows KDC with secure NFS 44

L

Linux Kerberos KDC, configure for secure NFS 54

M

managing 74
map file
 copying 76
 creating 76
mode bits 21

- mounting 33
 - file systems 33
- multihomed Data Movers 60

N

- naming services 30
- nas_automountmap 77
- netd file 28
- NFS
 - file locking 24
 - authentication for PC clients 20
 - creating a mount point 32
 - defined 11
 - disabling access 75
 - displaying statistics 97
 - exporting a file system 37
 - file locking 24
 - internationalization 22
 - managing 74
 - managing secure NFS 89
 - mounting a file system 33
- NFSv4 21, 22, 23, 25, 28, 30, 31, 32, 34, 36, 37, 82, 83, 89, 96, 97, 105
 - access control lists 21
 - client contexts 25
 - creating a mount point 32
 - delegation 23
 - displaying statistics 97
 - error messages 105
 - exporting a file system 36, 37
 - managing 82
 - managing secure NFS 89
 - mounting a file system 34
 - naming services 30
 - parameters 83, 89
 - 32bitClient 89
 - recallTimeout 83
 - setup procedures 28, 36
 - troubleshooting 96
 - Unicode 31
- NIS
 - configuring for secure NFS 44, 55

P

- parameters
 - 32bitClient 89
 - acl.sortAces 53
 - recallTimeout 83
 - secureExportMode 111
 - updatePTRrecord 44
- principals
 - adding service principals 61

- principals (*continued*)
 - mapping to UIDs 64
- protocol
 - NFS 11

R

- recallTimeout parameter 83
- reexporting 75
- reexporting file systems 75

S

- secure NFS 42, 44, 54, 95
 - troubleshooting 95
 - using a UNIX or Linux Kerberos KDC 54
 - using a Windows Kerberos KDC 44
- secureExportMode parameter 111
- security
 - Kerberos based 19
 - UNIX based 19
- server_export 37, 38, 40, 41, 42
 - access options 38
 - Kerberos 42
 - Kerberos security 42
 - NFSv4
 - access by NFSv4 clients only 41
 - setting access levels 38
 - UNIX security 42
 - VLAN access 40
- server_kerberos
 - creating realm 56
 - creating service principals 61
- server_mountpoint 32
- server_mountt 33
- server_name 55
- server_nfs
 - deleting principals 93
 - displaying mapping entry 92
 - displaying NFSv4 clients 85
 - displaying NFSv4 status 96
 - displaying user attributes 91
 - mapping Kerberos user principal names to UIDs 52
 - releasing authentication 94
 - releasing NFSv4 clients 88
 - setting the secure NFS service instance 49, 57
 - starting NFSv4 85
 - stopping NFSv4 84
- server_umount 35
- service principals
 - adding using the KDC 62

Solaris Kerberos KDC, configuring for secure NFS
Linux Kerberos KDC, configuring for secure NFS 54
system access 19
system access control 19

T

time services, configuring for secure NFS 44, 55

U

UIDs, mapping to user principal names 64

unexporting 75
unexporting, file systems 74
Unicode 31
UNIX security 19
unmounting 35
updatePTRrecord parameter 44
user authentication 19
user mapping 52
UTF-8 22

V

VLAN, specifying access by 40

