EMC® VNX® Series

Release 8.1

Managing Volumes and File Systems on VNX® Manually

P/N 300-015-131 Rev 01

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

Published August 2013

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.

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

Corporate Headquarters: Hopkinton, MA 01748-9103

Contents

Preface..7

Chapter 1: Introduction..9

Overview..10

System requirements..10

Restrictions..10

Cautions..11

User interface choices..12

Related information..12

Chapter 2: Concepts..15

Overview..16

File systems..16

Inode..17

Monitoring and repairing file systems...............................17

Volumes...18

Disk volumes..19

Slice volumes..19

Stripe volumes..20

Metavolumes...20

BCV...23

Planning considerations...25

Supported file system access protocols.............................25

File system size guidelines..26

NMFS..26

Volume configuration guidelines.....................................27

Stripe volume configuration considerations.....................28

Integration considerations.....28

Chapter 3: Configuring Volumes.....31
Determine storage availability.....32
Create volumes.....33
List volumes.....33
Create a slice volume.....34
Create a stripe volume.....34
Create a metavolume.....35
Provide storage from a VNX or legacy CLARiiON system to a gateway system.....36
RAID groups and storage characteristics 38
Add disk volumes to an integrated system.....39

Chapter 4: Configuring File Systems.....41
Create a file system.....42
(Optional) Create a mount point.....43
Mount a file system.....44
(Optional) Create an NMFS.....45
Create a new component file system.....46

Chapter 5: Managing Volumes.....49
Check volume capacity.....50
Rename a volume.....50
Clone a volume.....51
Delete a metavolume or stripe volume.....52
List metavolume information.....52
Delete a metavolume or stripe volume.....53
Delete a slice volume.....54
List slice volume information.....54
Delete a slice volume.....55

Chapter 6: Managing File Systems.....57
Export a file system.....58
Export a file system from a Data Mover for NFS access.....58
Export a file system from a Data Mover for CIFS access.....59
Export an NMFS.....59
List file systems.....60
View the file system configuration information.....61

List mount points....61
List mounted file systems....62
Check disk space capacity for a single mounted file system....63
Check disk space capacity for all mounted file systems....63
Check inode capacity....64
View the inode capacity of a single file system on a Data Mover....64
View the inode capacity of all file systems on a Data Mover....65
Extend a file system....66
Extend replicated file systems....67
Adjust file system size threshold....70
Adjust the file system size threshold for all file systems....70
Adjust the file system size threshold on a single Data Mover....71
Add an existing file system to an NMFS....71
Move an NMFS....72
Rename a file system....73
Enhance file read/write performance....74
Turn off read prefetch for a specific file system....74
Turn off read prefetch for all file systems on a Data Mover....75
Enable uncached write mechanism....76
Unmount all file systems from a Data Mover....76
Unmount all file systems temporarily....77
Unmount all file systems permanently....77
Delete a file system or NMFS....78

Chapter 7: Monitoring and Repairing File Systems....81
Run a file system check....82
Start an ACL check on the file system....82
List file system checks....83
Display the file system check information on a file system....83
Display information on all current file system checks....84

Chapter 8: Troubleshooting....85
EMC E-Lab Interoperability Navigator....86
VNX user customized documentation....86
Known problems and limitations....86
Error messages....87
EMC Training and Professional Services....87

Appendix A: GID Support...89

Overview...90

Restrictions for GID support...90

Glossary...91

Index..95

Preface

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

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

Special notice conventions

EMC uses the following conventions for special notices:

Note: Emphasizes content that is of exceptional importance or interest but does not relate to personal injury or business/data loss.

NOTICE Identifies content that warns of potential business or data loss.

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

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

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

Where to get help

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

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

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

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

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

Your comments

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

Please send your opinion of this document to:

`techpubcomments@EMC.com`

1

Introduction

Topics included are:

- Overview on page 10
- System requirements on page 10
- Restrictions on page 10
- Cautions on page 11
- User interface choices on page 12
- Related information on page 12

Overview

The EMC® VNX® system allows you to create and manage VNX volumes and file systems manually or automatically.

This document explains the manual process for creating, configuring, and managing volumes and file systems, including:

- Creating and managing nested mount file systems (NMFS)
- Performing basic file system administration tasks

Managing Volumes and File Systems with VNX AVM provides information on configuring the VNX system to perform these tasks by using Automatic Volume Management (AVM) storage pools.

This document is part of the VNX documentation set and is intended for use by system administrators responsible for creating and managing VNX volumes and file systems manually.

System requirements

Table 1 on page 10 describes the EMC VNX series software, hardware, network, and storage configurations.

Table 1. System requirements

Software	VNX series version 8.1
Hardware	No specific hardware requirements
Network	No specific network requirements
Storage	Any VNX-qualified storage system

Restrictions

- When creating volumes on a VNX system attached to an EMC Symmetrix® storage system, use regular Symmetrix volumes (also called hypervolumes), not Symmetrix metavolumes.
- You must use LUNs that have the same data service policies. If you mix LUNs, you will receive a warning during diskmark.

Cautions

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

- EMC does not recommend spanning a file system (including checkpoint file systems) across multiple storage systems. All parts of a file system must use the same type of disk storage and be stored on a single storage system. Spanning more than one storage system increases the chance of data loss or data unavailability or both. This is primarily due to the high-availability concern because one storage system could fail while the other continues, making failover difficult. In this case, the targets might not be consistent. In addition, a spanned file system is subject to any performance and feature set differences between storage systems.
- Too many files in the root (/) of any file system might impact system performance. For optimal performance, the number of objects (such as files and subdirectories) should not exceed 500 names.
- Review Integration considerations on page 28 if you intend to use file systems with VNX features such as international character sets, EMC SnapSure™, EMC TimeFinder®/FS, quotas, or an antivirus agent (VEE CAVA).
- If you plan to set quotas on a file system to control the amount of space that users and groups can consume, turn on quotas immediately after creating the file system. Turning on quotas later, when the file system is in use, can cause temporary file system disruption, including slow file system access, for systems that use version 6.0.40 and earlier, and can impact system performance for systems that use version 6.0.41 and later. *Using Quotas on VNX* contains instructions on turning on quotas and general quotas information.
- If your user environment requires international character support (that is, support of non-English character sets or Unicode characters), configure the system to support this feature before creating file systems. *Using International Character Sets on VNX for File* contains instructions to support and configure international character support.
- If you plan to create TimeFinder/FS (local, NearCopy, or FarCopy) snapshots, do not use slice volumes (nas_slice) when creating the production file system (PFS). Instead, use the full portion of the disk presented to the system. Using slice volumes for a PFS slated as the source for snapshots wastes storage space and can result in loss of PFS data.
- Do not attempt to use Symmetrix TimeFinder tools and utilities with file system copies created by VNX TimeFinder/FS. It might result in loss of data.
- Do not manually edit the nas_db database without consulting EMC Customer Support. Any changes to this database might cause problems when installing the system.
- Permanently unmounting all file systems from a Data Mover must be done with caution because this operation deletes the contents of the mount table. To reestablish client access to the file systems after this operation, rebuild the mount table by remounting each file system on the Data Mover.

- The file system is unavailable to users during a file system check (fsck). NFS clients receive an "NFS server not responding" message. CIFS clients lose the server connection and must remap shares.
- Depending on the file system size, the fsck utility might use a significant amount of the system's resources (memory and CPU) and might affect overall system performance.
- Only two fsck processes can run on a single Data Mover simultaneously.
- A file system check of a permanently unmounted file system can be executed on a standby Data Mover.
- If a Data Mover restarts or experiences failover or failback while running the fsck utility on an unmounted file system, restart the fsck utility on the unmounted file system.

User interface choices

The system offers flexibility in managing networked storage based on your support environment and interface preferences. This document describes how to create and manage file systems and their underlying volumes by using the command line interface (CLI). You can also perform many of these tasks by using one of the following management applications:

- EMC Unisphere® software
- Celerra Monitor
- Microsoft Management Console (MMC) snap-ins
- Active Directory Users and Computers (ADUC) extensions

The Unisphere software online help contains additional information about managing your system.

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

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

Related information

Specific information related to the features and functionality described in this document is included in:

- *VNX Command Line Interface Reference for File*
- VNX for File man pages
- *Parameters Guide for VNX for File*
- *VNX Glossary*
- *Configuring and Managing CIFS on VNX*

- *Configuring NFS on VNX*
- *Managing a Multiprotocol Environment on VNX*

Volumes and file systems information that is related, but is beyond the scope of this document, is included in *Managing Volumes and File Systems with VNX AVM*.

Other related EMC publications include *Using FTP and TFTP on VNX*.

EMC VNX documentation on EMC Online Support

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

VNX wizards

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

2

Concepts

Topics included are:

- Overview on page 16
- File systems on page 16
- Volumes on page 18
- Planning considerations on page 25

Overview

The system offers flexible volumes and file systems management.

Manual volume management allows you to create and aggregate different volume types into usable file system storage that meets your configuration needs. When you create and manage volumes manually, you have greater control over the location of storage allocated to a file system. There are a variety of volume types and configurations from which you can choose to optimize your file system's storage potential. You can divide, combine, and group volumes to meet your specific configuration needs.

You can also manage VNX volumes and file systems without having to create and manage underlying volumes. AVM is a feature that automatically creates and manages usable file system storage. Although AVM is a simple way to create volumes and file systems, automation can limit your control over the location of the storage allocated to a file system. *Managing Volumes and File Systems with VNX AVM* provides additional information on AVM capabilities of the system.

File systems

A file system is a method of naming and logically organizing files and directories on a storage system. A VNX file system must be created and stored on a metavolume. The metavolume provides:

- Expandable storage capacity that might be needed to dynamically expand a file system
- The means to form a logical volume that is larger than a single disk

A metavolume can include disk volumes, slice volumes, stripe volumes, or other metavolumes.

The VNX system creates different file systems based on how they are used. Table 2 on page 16 lists the types of file system.

Table 2. File system types

File system type	ID	Description
uxfs	1	Default file system
rawfs	5	Raw file system
mirrorfs	6	Mirrored file system
ckpt	7	Checkpoint file system
mgfs	8	Migration file system
group	100	Members of a file system group

Table 2. File system types *(continued)*

File system type	ID	Description
vpfs	--	Volume pool file system (multiple services sharing a single metavolume)
rvfs	--	Local configuration volume used to store replication-specific internal information
nmfs	102	Nested mount file systems (NMFS)

Inode

An inode is a data structure that stores information on files and the location of file blocks in the NFS file system. The VNX system uses this information to identify if the file is a regular file, a directory, or a symbolic link.

Each file requires at least one inode. Without inodes, you cannot create any new files, even if there is space on the hard drive. The number of bytes per inode (nbpi) specifies the density of inodes in the file system. EMC recommends an nbpi value of one inode for every 8,192 bytes, which is the default setting.

Monitoring and repairing file systems

Occasionally, a file system might get corrupted if the system is shut down improperly or the disk suffers a minor failure. In these situations, it might be necessary to try to repair the file system by using the fsck utility. Cautions on page 11 provides information on file system behavior during the fsck process.

The fsck utility checks file system consistency on a file system by detecting and correcting file system storage errors.

When a file system corruption is detected during runtime, the Data Mover panics and the restart or failover process starts.

During the restart process, file systems found to be corrupted are not mounted. Run the nas_fsck command manually on these file systems during a suitable time window. You can also use the nas_fsck command to check the status through the Control Station.

When the ufs.skipFsck parameter is set to True (default), the restart process does not run fsck and the corrupted file systems are not mounted. To override this behavior, set this parameter to False. The *Parameters Guide for VNX for File* provides detailed information on parameters.

When the system begins fsck on a mounted file system, the fsck utility automatically unmounts the file system, runs fsck, and then remounts the file system. The file system is unavailable for the duration of the fsck. NFS clients see an "NFS server not responding" message. CIFS clients lose connectivity to the server and must remap shares.

The fsck utility should not be run on a server under heavy load to prevent the server from running out of resources. In most cases, the user is notified when sufficient memory is unavailable to run fsck. In these cases, users can choose one of these options:

- Start fsck during off-peak hours.
- Restart the server and start fsck immediately.
- Run fsck on a different server if the file system is unmounted.

The fsck utility cannot run on a read-only file system. You do not need to run fsck for normal restart or shutdown operations. File system consistency is maintained through a logging mechanism and restart and shutdown operations cause no corruption.

The first step in the fsck process is to ensure that the corruption can be safely corrected without bringing down the server. The fsck process also corrects any inconsistencies in the Access Control List (ACL) database. The corrupted file system is unavailable to users during the fsck process. After the fsck utility finds and corrects the corruption, users regain access to the file system. While fsck is running, other file systems mounted on the same server are not affected and are available to users.

Volumes

A volume is a virtual disk into which a file system places data. Volumes underlie the file system. Create volumes and metavolumes to assign storage space for a file system.

The VNX system supports the volume types listed in Table 3 on page 18. Each volume type provides different benefits to satisfy storage requirements. With the exception of system volumes, all volumes are initially available for data storage.

Table 3. Volume types

Volume type	Description
Disk volume	Represents the basic disk configuration on the storage system.
Slice volume	Typically provides a smaller disk section.
Stripe volume	Provides an arrangement of volumes organized into a set of interlaced stripes that improve volume performance.
Metavolume	Provides a concatenation of slice, stripe, disk, or metavolumes. Metavolumes are required for dynamic file system storage.
Business continuance volume (BCV)	Enables the use of TimeFinder/FS file system copies. BCV is only supported on a VNX system attached to a Symmetrix system.

Volume configuration guidelines on page 27 provides additional information on common volume configurations and considerations associated with each volume type.

Disk volumes

A disk volume is a physical storage unit that is exported from the storage system to the VNX for File. Disk volumes are the underlying storage of all other volume types.

A disk volume equates to a LUN as presented to the VNX for File by the storage system. Each LUN is a usable storage-system volume that appears as a disk volume to the VNX system. Disk volumes are typically created by EMC support personnel during the initial installation and setup of the VNX system. After the initial installation and setup, configure disk volumes only when you add LUNs to the storage system.

Slice volumes

A slice volume is a logical, nonoverlapping section cut from another volume component. When you create a slice volume, you can indicate an offset. The offset is the distance in megabytes between the disk and the start of a slice volume. If an offset is not specified, the system places the slice in the first-fit algorithm (default), that is, the next available volume space. An offset is rarely specified.

You must first identify the volume from which the slice volume will be created. The root slice volumes that are created during installation appear when you list the volume configurations. However, to protect the system, you do not have access privileges to them, and therefore, cannot execute any commands against them.

Slice volumes can be configured to any size, but are typically used to create smaller, more manageable units of storage. The definition of a "more manageable" logical volume size depends on the system configuration and the type of data you are storing. Slicing is more common with EMC VNX for Block storage systems because of the larger LUNs presented to the VNX for File.

Figure 1 on page 19 shows an 18 GB volume on which a 2 GB slice is defined.

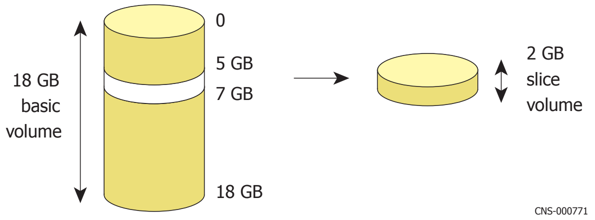

Figure 1. Slice volumes

A disk volume, stripe volume, or metavolume used as part of a business continuance volume (BCV) should not be sliced. BCV on page 23 provides information on using BCVs.

You must configure slice volumes as part of a metavolume to store file system data on them. Metavolumes on page 20 provides additional information.

Stripe volumes

A stripe volume is a logical arrangement of participating disk volumes, slice volumes, or metavolumes organized, as equally as possible, into a set of interlaced stripes. Stripe volumes achieve greater performance and higher aggregate throughput because all participating volumes can be active concurrently.

Figure 2 on page 20 shows an example of a stripe volume. The stripe is created across four participating volumes of equal size.

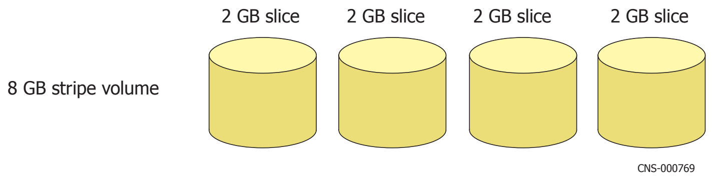

Figure 2. Stripe volumes

Stripe volumes improve performance because, unlike disk volumes, slice volumes, and metavolumes, addressing within a stripe volume is conducted in an interlaced fashion across volumes, rather than sequentially.

In a stripe volume, a read request is spread across all component volumes concurrently. Figure 3 on page 20 shows addressing within a stripe volume.

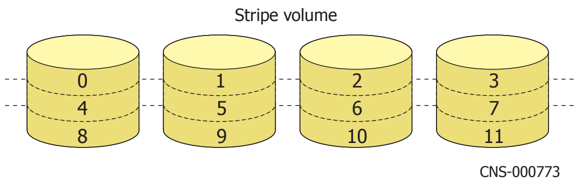

Figure 3. Addressing a stripe volume

Data is interlaced within the stripe volume starting with stripe unit 0 on the first participating volume, continuing to stripe unit 1 on the next participating volume, and so on. As necessary, data wraps back to the first participating volume. This is controlled by stripe depth, which is the amount of data written to a participating stripe volume member before moving on to the next participating member. Stripe volume configuration considerations on page 28 provides guidelines to configure a stripe volume.

Metavolumes

File systems can only be created and stored on metavolumes. A metavolume is an end-to-end concatenation of one or more disk volumes, slice volumes, stripe volumes, or metavolumes.

A metavolume is required to create a file system because metavolumes provide the expandable storage capacity needed to dynamically expand file systems. A metavolume also provides a way to form a logical volume larger than a single disk.

EMC recommends using only one type of volume within a metavolume. For example, you can create and combine stripe volumes, define them as a metavolume, and then create a file system with space from that metavolume.

Configuring a metavolume

Metavolumes can be created from a disk volume, stripe volume, slice volume, or another metavolume. A file system is created on the metavolume.

You can expand a metavolume by adding additional disk volumes, stripe volumes, slice volumes, or metavolumes to it.

When you extend a file system that is on a metavolume, the metavolume is automatically extended. Figure 4 on page 21 shows a metavolume configuration that uses three disk volumes.

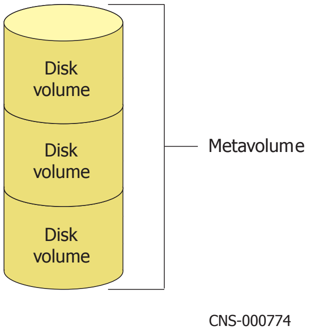

Figure 4. Metavolume configuration

Addressing within a metavolume

All information stored within a metavolume is arranged in addressable logical blocks and is organized in a sequential, end-to-end fashion. Figure 5 on page 22 shows metavolume addressing.

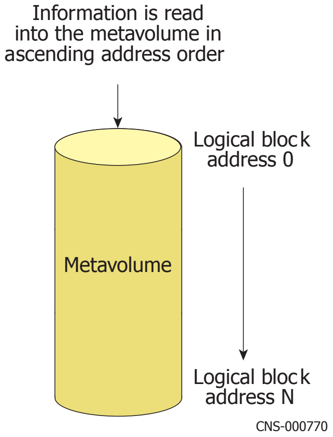

Figure 5. Metavolume addressing

Metavolumes can be created by using slice and stripe volumes.

You can create a 12 GB metavolume on four disk volumes by creating 3 GB slices on each disk. You can then create your file system on the metavolume as shown in Figure 6 on page 22.

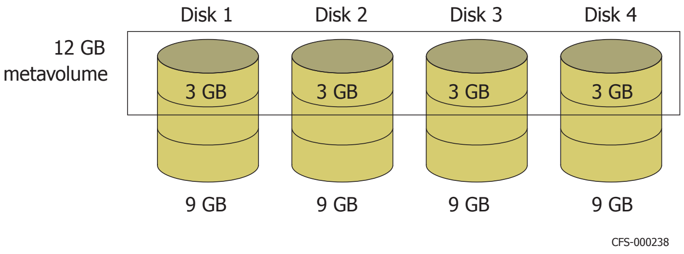

Figure 6. Metavolume created from slice volumes

Create striped volumes over a specified number of disk volumes by defining a 256 KB stripe depth. Put these striped volumes together to create a striped metavolume. You can then create a file system on the metavolume as shown in Figure 7 on page 23.

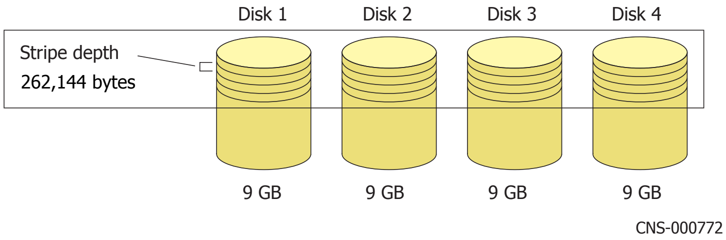

Figure 7. Metavolume created from stripe volumes

Note: The total capacity of a metavolume equals the sum of all volumes that compose the metavolume.

BCV

BCVs are dedicated volumes that can be attached to a standard volume on which a file system resides. The TimeFinder/FS feature of the VNX system uses BCVs to create file system copies and mirror file systems dynamically. The EMC Customer Support Representative creates BCVs on the storage system before installing the VNX software.

Note: BCVs are supported only for Symmetrix storage systems.

When planning for BCVs, ensure that you have as many BCVs as standard disk volumes to be used by the largest file system. Figure 8 on page 23 shows the relationship between standard volumes and BCVs.

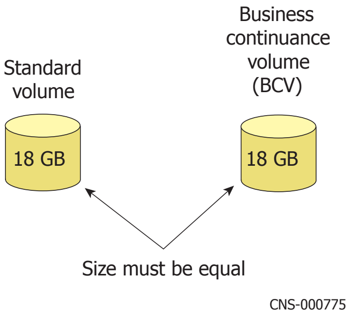

Figure 8. BCV

Note: In Figure 8 on page 23, size refers to a disk volume's geometry.

BCVs are based on the LUNs (entire disk volumes) that are presented to the VNX system. BCVs should not use slice volumes because TimeFinder/FS operations are run against the entire disk volume. Disk volumes, stripe volumes, and metavolumes used in BCVs should not be sliced.

The TimeFinder/FS feature uses BCVs to create file system copies and mirror file systems dynamically:

- With the file system copy function, you can create an exact copy of a file system to use as input to a backup or restore operation, for application development, or for testing.
- With the mirror function, you can create a file system copy in which all changes to the original file system are reflected in the mirrored file system.

After a BCV is created, use the fs_timefinder command to create a file system copy.

NOTICE Do not attempt to use Symmetrix TimeFinder tools and utilities with file system copies created by VNX TimeFinder/FS. It might result in loss of data.

Planning considerations

This section provides information that is helpful to plan file systems for the VNX system:

- Supported file system access protocols on page 25
- File system size guidelines on page 26
- NMFS on page 26
- Volume configuration guidelines on page 27
- Stripe volume configuration considerations on page 28
- Integration considerations on page 28

Supported file system access protocols

The VNX system supports access to file systems by using NFS, CIFS, FTP, and TFTP protocols. Because the protocols used in your environment affect how users access file systems, it is important that you understand the types of users in your environment. The underlying volume structure for your file systems should support the protocols you use and allow users to access file systems in the normal manner.

The following documents contain more information on setting up and managing access to a file system by using a particular protocol:

- *Configuring NFS on VNX*
- *Configuring and Managing CIFS on VNX*
- *Managing a Multiprotocol Environment on VNX*

  The VNX system lets you map Windows users and groups to UNIX user IDs (UIDs) and group IDs (GIDs) to provide users with seamless access to shared file system data. *Configuring VNX User Mapping* provides additional information.

- *Using FTP and TFTP on VNX*

  FTP is a client/server protocol that operates over TCP/IP and allows file uploading and downloading across heterogeneous systems. FTP includes functions to log in to the remote system, list directories, and copy files.

  A Trivial File Transfer Protocol (TFTP) is a simple, UDP-based protocol to read and write files. TFTP can be used to boot remote clients from a network server. TFTP does not authenticate users or provide access control.

File system size guidelines

The size of file systems depends on a variety of factors within a business organization. Follow these general guidelines when you create file systems in your environment:

- Consider the size of the PFS and plan the backup and restore strategy. Larger file systems might take more time to back up and restore. If a file system becomes inconsistent, larger file systems might take longer to run a consistency check.
- Consider the rate at which the data grows and plan to increase the space as the data increases.
- On each Data Mover, the total size of all file systems, the size of all SavVols used by SnapSure, and the size of all SavVols used by the EMC VNX Replicator feature must be less than the total supported capacity of the Data Mover.

The EMC E-Lab™ Interoperability Navigator available on EMC Online Support contains more information on file system size guidelines.

NMFS

An NMFS allows you to manage a collection of component file systems as a single file system. CIFS and NFS clients see component file systems as a single share or single export.

File system capacity is managed independently for each component file system. This means that you can increase the total capacity of the NMFS by extending an existing component file system or by adding new component file systems.

The number of NMFS or component file systems is limited only to the number of file systems allowed on a Data Mover. Hard links (NFS), renames, and simple moves are not possible from one component file system to another.

The VNX features that support NMFS are:

- SnapSure
- VNX Replicator
- Quotas
- Security policies
- Backup
- EMC Symmetrix Remote Data Facility (SRDF®)
- TimeFinder/FS
- File system migration
- FileMover
- EMC MirrorView™/Synchronous

Volume configuration guidelines

Volume configuration guidelines are helpful to plan the storage structure of file systems. Table 4 on page 27 lists common volume configurations and considerations associated with each volume type.

Table 4. Volume configuration guidelines

Volume configurations	Considerations
disk—>stripe—>meta	Allows load balancing of disk subsystem. Spreads data across all disks. Efficient for random read operations. Can use with TimeFinder/FS and BCVs.
disk—>meta	Simpler to manage. Efficient for sequential reads and writes; for example, logging information. Some disks in the metavolume might be frequently accessed while others remain idle. Can use with TimeFinder/FS and BCVs.
disk—>stripe—>slice—>meta	Striping can lead to better performance. Slicing allows the creation of smaller file systems. Less complex volume structure. Can create file system by using only a part of volume and use the remaining volume space for other applications. Slicing is not recommended when using BCVs.
disk—>slice—>stripe—>meta	Slicing allows the creation of smaller file systems. Striping can lead to better performance. Can create file system by using only a part of the volume and use the remaining volume space for other applications. Slicing is not recommended when using BCVs.

Stripe volume configuration considerations

The following are guidelines to configure a stripe volume:

- The use of different stripe sizes depends on the applications you are using. The stripe depth must be specified in multiples of 8,192 bytes. EMC recommends a stripe size of 262,144 bytes (256 KB), which is the default, for file systems that run in a CIFS or NFS environment with a Symmetrix or VNX for Block storage system.
- Consider the size of the stripe volume. After the stripe volume is created, its size remains fixed. However, you can extend a file system built on top of a stripe volume by combining or concatenating it with additional stripe volumes.
- For optimal performance, stripe across different volumes. While striping across a single volume is possible, it does not improve performance.
- Configure stripes as follows to use the maximum amount of disk space:
  - The size of the participating volumes within the stripe should be uniform and evenly divisible by the size of the stripe.
  - Each participating volume should contain the same number of stripes.
  - Space is wasted if the volumes are evenly divisible by the stripe size but are unequal in capacity. The residual space is not included in the configuration and is unavailable for data storage.
- If eight or more volumes are available, building stripe volumes on multiples of eight volumes should give reasonable performance in most environments. If eight volumes do not provide sufficient file system capacity, combine as many sets of eight volumes as necessary into a single metavolume.

Integration considerations

This section identifies considerations for successful file system operations and integration when using:

- Quotas on page 29
- TimeFinder/FS on page 29
- File-level retention on page 29
- SRDF on page 30
- VNX Replicator/SnapSure on page 30
- MirrorView/Synchronous on page 30

Quotas

To ensure that file systems do not become full, you can impose quota limits on users and groups that create files on directory trees. You can set a hard quota limit on user, group, or directory tree quotas to prevent allocation of all the space in the file system. When the hard quota limit is reached, the system denies user requests to save additional files and notifies the administrator that the hard quota limit has been reached. In this case, existing files can be read but action must be taken either by the user or administrator to delete files from the file system or increase the hard quota limit to allow saving of additional files.

To avoid degradation of file system performance, set the hard quota limit between 80 and 85 percent of the total file system space. In addition to setting the hard quota limit, set a lower soft quota limit so that the administrator is notified when the hard quota limit is being approached.

For example, to prevent a file system that contains 100 GB of storage from filling up, you can set a soft quota limit of 80 GB and a hard quota limit of 85 GB by using user, group, or directory tree quotas. When used space in the file system reaches 80 GB, the administrator is notified that the soft limit is reached. When used space totals 85 GB, the system denies user requests to save additional files, and the administrator is notified that the hard quota limit is reached.

Using Quotas on VNX provides detailed information on quotas and how to set up user, group, or directory tree quotas.

TimeFinder/FS

If you plan to create multiple copies of your PFS, plan for that number of BCVs. For example, from one PFS, you can create 10 copies. Therefore, plan for 10 BCVs, not one.

TimeFinder/FS uses the physical disk, not the logical volume, when it creates BCV copies. The copy is done track-by-track, so unused capacity is carried over to the BCVs.

Volumes used for BCVs should be of the same size as the standard volume.

Using TimeFinder/FS, NearCopy, and FarCopy on VNX provides additional information on TimeFinder/FS.

File-level retention

File systems can only be enabled with file-level retention (FLR) capability at creation time. When the file system is created and enabled for FLR, it is persistently marked as an FLR file system and the FLR setting cannot be changed. After a file system is created and FLR is enabled, an administrator can apply FLR protection to individual files. Files in the FLR (locked) state can be stored with retention dates, which prohibit the deletion of the file until expiration.

Using VNX File-Level Retention provides more information about FLR storage and FLR file system behavior.

SRDF

All file systems on the Data Mover must be built on SRDF volumes. *Using SRDF/S with VNX for Disaster Recovery* describes SRDF/S and *Using SRDF/A with VNX* describes SRDF/A.

If you use the AVM feature to create the file systems, specify the symm_std_rdf_src storage pool. This storage pool directs AVM to allocate space from volumes configured during installation for remote mirroring by using SRDF.

Automatic file system extension cannot be used for any file system that is part of a Remote Data Facility (RDF) configuration.

Note: Do not use the nas_fs command with the -auto_extend option for file systems associated with RDF configurations. Doing so generates the error message: Error 4121: operation not supported for file systems of type SRDF.

VNX Replicator/SnapSure

By using VNX Replicator, you can enable automatic file system extension on the source file system. When the source file system hits its high water mark (HWM), the destination file system automatically extends first. Then the source file system automatically extends. If the extension of the destination file system succeeds but the source file system extension fails, the file systems differ in size, which causes replication failure. Use the nas_fs -xtend <fs_name> -option src_only command to manually adjust the size of the source file system. *Using VNX Replicator* contains instructions to recover from this situation. *Managing Volumes and File Systems with VNX AVM* provides information on automatic file system extension.

There must be sufficient file system space and disk storage available to support VNX Replicator and SnapSure operations. To review the entire file system size, use the nas_fs -list command. To calculate the SavVol file size, use the nas_disk -size command. The *VNX Command Line Interface Reference for File* provides a detailed synopsis of the commands associated with SnapSure and VNX Replicator.

MirrorView/Synchronous

All file systems on the Data Mover must be built on MirrorView/Synchronous LUNs. *Using MirrorView/Synchronous with VNX for Disaster Recovery* provides detailed information on MirrorView/Synchronous.

If you use the AVM feature to create the file systems, you must use the appropriate MirrorView AVM storage pools for your RAID configuration. *Managing Volumes and File Systems with VNX AVM* provides a list of the MirrorView AVM storage pools.

3

Configuring Volumes

The tasks to configure volumes manually are:

- Determine storage availability on page 32
- Create volumes on page 33
- Provide storage from a VNX or legacy CLARiiON system to a gateway system on page 36
- Add disk volumes to an integrated system on page 39

Determine storage availability

When reserving disk volumes for manual volume management or for use with AVM user-defined storage pools, reserve disk volumes in SP-balanced pairs of the same RAID type, disk count, and size. Although you can use all available disk volumes with AVM user-defined storage pools, its performance might be impacted if the volumes are not configured this way.

Before you create a new volume, identify the unused disk space. If a disk is unused, its space is available for volume and file system creation. Determine storage availability on the system by identifying the unused disks.

Action

To view a list of unused disks and their sizes, type:

```
$ nas_disk -list
```

Output

```
id inuse sizeMB  storageID-devID   type  name        servers
1   y     4153    000183501491-000  STD   root_disk   1,2,3,4
2   y     4153    000183501491-001  STD   root_ldisk  1,2,3,4
3   y     8631    000183501491-00C  STD   d3          1,2,3,4
4   y     8631    000183501491-00D  STD   d4          1,2,3,4
5   y     8631    000183501491-00E  STD   d5          1,2,3,4
6   y     8631    000183501491-00F  STD   d6          1,2,3,4
7   n     8631    000183501491-010  STD   d7          1,2,3,4
8   n     8631    000183501491-011  STD   d8          1,2,3,4
9   n     8631    000183501491-012  STD   d9          1,2,3,4
10  n     8631    000183501491-013  STD   d10         1,2,3,4
11  n     8631    000183501491-014  STD   d11         1,2,3,4
```

Note

Column definitions:

id — ID of the disk (assigned automatically)

inuse — Indicates whether the disk is in use by a file system

sizeMB — Size of disk in megabytes

storageID-devID — ID of the storage system and device associated with the disk

type — Type of disk

name — Name of the disk

servers — Data Movers with access to the disk

Create volumes

You can create three types of volumes:

- Slice volume
- Stripe volume
- Metavolume

Each volume type provides different benefits to satisfy storage requirements. Volumes on page 18 provides detailed information on volume types. Volume configuration guidelines on page 27 lists common volume configurations and considerations associated with each volume type.

List volumes

Action

To list all volumes on a system, type:

```
$ nas_volume -list
```

Output

This is a partial listing of the volume table that is displayed as the volume list:

```
id inuse type acl name       cltype clid
1  y     4    0   root_disk  0      1,2,3,4,5,6,7,8,9,10,11,12,13,
                                     14,15,16,17,18,19,20,21,22,
                                     23,24,25,26,27,28,29,30,31,
                                     32,33,34,35,36,37,38,39,40,
                                     41,42,43,44,45,46,47,48,49,
                                     50
2  y     4    0   root_ldisk 0      51,52,53,54,55,56,57,58,59,60,
                                     61,62,63,64,65,66
3  y     4    0   d3         1      246
4  y     4    0   d4         1      246
5  y     4    0   d5         1      246
6  y     4    0   d6         1      246
7  y     4    0   d7         0      67,68
8  y     4    0   d9         1      248
10 y     4    0   d10        1      257
11 y     4    0   d11        1      249
12 y     4    0   d12        1      249
13 y     4    0   d13        1      249
14 y     4    0   d14        1      249
15 y     4    0   d15        1      257
.
.
.
246 y    2    0   stv1       1      247
247 n    3    0   mtv1       0
248 n    4    0   mtv2       0
```

Note

You can also use the nas_slice -list command to list only slice volumes.

Column definitions:

id — ID of the volume (assigned automatically)

inuse — Indicates whether the volume is in use by a file system; y indicates yes, n indicates no

type — Type of volume

acl — Access control value assigned to the volume

name — Name of the volume

Create a slice volume

Action

To create a slice volume, use this command syntax:

```
$ nas_slice -name <name> -create <volume_name> <size>
```

where:

<name> = name of the slice volume

<volume_name> = name of the volume

<size> = size of the slice volume in megabytes

Example:

To create a slice volume named slv1, type:

```
$ nas_slice -name slv1 -create d8 1024
```

Output

```
id          = 76
name        = slv1
acl         = 0
in_use      = False
slice_of    = d8
offset(MB)  = 0
size   (MB) = 1024
volume_name = slv1
```

Create a stripe volume

When creating a stripe volume, if you do not specify a name for the stripe volume, a default name is assigned. Stripe volume configuration considerations on page 28 provides more information.

Action

To create a stripe volume, use this command syntax:

```
$ nas_volume -name <name> -create -Stripe <stripe_size> [<volume_name>,...]
```

where:

<name> = name of the stripe volume

<stripe_size> = stripe element size, in bytes (must be a multiple of 8,192 bytes)

<volume_name> = names of the volumes separated by commas

Example:

To create a stripe volume called stv1, type:

```
$ nas_volume -name stv1 -create -Stripe 32768 d10,d12,d13,d15
```

Output

```
id         = 125
name       = slv1
acl        = 0
in_use     = False
type       = stripe
volume_set = d10,d12,d13,d15
disks      = d10,d12,d13,d15
```

Create a metavolume

When creating a metavolume, if you do not specify a name for the metavolume, a default name is assigned.

To combine volumes into a metavolume, use the <volume_name> option consecutively in the command syntax.

Action

To create a metavolume from a stripe volume, use this command syntax:

```
$ nas_volume -name <name> -create -Meta [<volume_name>,...]
```

where:

<name> = name of the stripe volume

<volume_name> = names of the metavolumes separated by commas

Example:

To create metavolumes named slv1, slv2, and slv3 on disk volume d7, type:

```
$ nas_volume -name mtv1 -create -Meta slv1,slv2,slv3
```

Output

```
id          = 268
name        = mtv1
acl         = 0
in_use      = False
type        = meta
volume_set  = slv1, slv2, slv3
disks       = d8, d19, d9
```

Provide storage from a VNX or legacy CLARiiON system to a gateway system

Before you begin

If a VNX for File gateway system is attached to a VNX for Block storage system and you want to add disk volumes to the configuration, use this procedure.

EMC VNX for File network servers that are gateway network-attached storage systems and that connect to EMC Symmetrix and VNX for Block storage systems are:

- VNX VG2
- VNX VG8

A VNX for File gateway system stores data on VNX for Block user LUNs or Symmetrix hypervolumes. If the user LUNs or hypervolumes are not configured correctly on the storage system, AVM and the Unisphere for File software cannot be used to manage the storage.

Typically, an EMC Customer Support Representative does the initial setup of disk volumes on these gateway storage systems.

Use the procedures that follow to add disk volumes to a configuration:

1. Use the Unisphere software or the VNX for Block CLI to create user LUNs.
2. Use either the Unisphere software or the VNX for File CLI to make the new user LUNs available to the VNX for File as disk volumes.

Create user LUNs before you create file systems.

To add user LUNs, you must be familiar with the following:

- Unisphere software or the VNX for Block CLI.
- Process of creating RAID groups and user LUNs for the VNX for File volumes.

The documentation for Unisphere and VNX for Block CLI, describes how to create RAID groups and user LUNs.

Procedure

1. Create RAID groups and user LUNs as needed for VNX for File volumes. Ensure that you add the LUNs to the VNX for File gateway system's storage group and that you set the HLU to 16 or higher:

   - Always create the user LUNs in balanced pairs, one owned by SP A and one owned by SP B. The paired LUNs must be the same size.
   - FC or SAS disks must be configured as RAID 1/0, RAID 5, or RAID 6. The paired LUNs do not need to be in the same RAID group but should be of the same RAID type. RAID groups and storage characteristics on page 38 lists the valid RAID group and storage system combinations. Gateway models use the same combinations as the NS-80 (for CX3™ series storage systems) or the NS-960 (for CX4™ series storage systems).
   - SATA disks must be configured as RAID 1/0, RAID 5, or RAID 6. All LUNs in a RAID group must belong to the same SP. Create pairs by using LUNs from two RAID groups. RAID groups and storage characteristics on page 38 lists the valid RAID group and storage system combinations. Gateway models use the same combinations as the NS-80 (for CX3 series storage systems) or the NS-960 (for CX4 series storage systems).
   - The host LUN identifier (HLU) must be greater than or equal to 16 for user LUNs.

   Use these settings when creating RAID group user LUNs:

   - RAID Type: RAID 1/0, RAID 5, or RAID 6 for FC or SAS disks and RAID 1/0, RAID 5, or RAID 6 for SATA disks
   - LUN ID: Select the first available value
   - Rebuild Priority: ASAP
   - Verify Priority: ASAP
   - Enable Read Cache: Selected
   - Enable Write Cache: Selected
   - Enable Auto Assign: Cleared (off)
   - Number of LUNs to Bind: 2
   - Alignment Offset: 0
   - LUN size: Must not exceed 14 TB

2. Perform one of these steps to make the new user LUNs available to the VNX for File:

   - Using the Unisphere software:

     a. Select **Storage ➤ Storage Configuration ➤ File Systems**.

     b. From the task list, select **File Storage ➤ Rescan Storage Systems**.

   - Using the VNX for File CLI, type the following command:

     ```
     nas_diskmark -mark -all
     ```

Note: Do not change the HLU of the system LUNs after rescanning. This might cause data loss or data unavailability.

RAID groups and storage characteristics

The following table correlates the storage array to the RAID groups.

Table 5. RAID group combinations

Storage	RAID 5	RAID 6	RAID 1
NX4 SAS or SATA	2+1 RAID 5 3+1 RAID 5 4+1 RAID 5 5+1 RAID 5	4+2 RAID 6	1+1 RAID 1/0
NS20 / NS40 / NS80 FC	4+1 RAID 5 8+1 RAID 5	4+2 RAID 6 6+2 RAID 6 12+2 RAID 6	1+1 RAID 1
NS20 / NS40 / NS80 ATA	4+1 RAID 5 6+1 RAID 5 8+1 RAID 5	4+2 RAID 6 6+2 RAID 6 12+2 RAID 6	Not supported
NS-120 / NS-480 / NS-960 FC	4+1 RAID 5 8+1 RAID 5	4+2 RAID 6 6+2 RAID 6 12+2 RAID 6	1+1 RAID 1/0
NS-120 / NS-480 / NS-960 ATA	4+1 RAID 5 6+1 RAID 5 8+1 RAID 5	4+2 RAID 6 6+2 RAID 6 12+2 RAID 6	1+1 RAID 1/0
NS-120 / NS-480 / NS-960 Flash	4+1 RAID 5 8+1 RAID 5	Not supported	1+1 RAID 1/0

Add disk volumes to an integrated system

Configure unused or new disk devices on a VNX for Block storage system by using the Disk Provisioning Wizard for File. This wizard is available only for integrated VNX for File models (non-gateway systems), including Fibre Channel-enabled models, attached to a single VNX for Block storage system.

Note: For VNX systems, Advanced Data Service Policy features such as FAST VP and compression are supported on pool-based LUNs only. They are not supported on RAID-based LUNs.

To open the Disk Provisioning Wizard for File in the Unisphere software:

1. Select **Storage ➤ Storage Configuration ➤ Storage Pools**.
2. From the task list, select **Wizards ➤ Disk Provisioning Wizard for File**.

Note: To use the Disk Provisioning Wizard for File, you must log in to the Unisphere software by using the global sysadmin user account or by using a user account which has privileges to manage storage.

An alternative to the Disk Provisioning Wizard for File is available by using the VNX for File CLI at /nas/sbin/setup_clariion. This alternative is not available for unified VNX systems. The script performs the following actions:

- Provisions the disks on integrated (non-Performance) VNX for Block storage systems when there are unbound disks to configure. This script binds the data LUNs on the DPEs and DAEs, and makes them accessible to the Data Movers.
- Ensures that your RAID groups and LUN settings are appropriate for your VNX for File server configuration.

VNX systems must use the User_Defined mode with the /nas/sbin/setup_clariion CLI script.

The setup_clariion script allows you to configure VNX for Block storage systems on a shelf-by-shelf basis by using predefined configuration templates. For each enclosure (DPE or DAE), the script examines your specific hardware configuration and gives you a choice of appropriate templates. You can mix combinations of RAID configurations on the same storage system. The script then combines the shelf templates into a custom, User_Defined array template for each VNX for Block system, and then configures your array.

4

Configuring File Systems

The tasks to configure file systems are:

- Create a file system on page 42
- (Optional) Create a mount point on page 43
- Mount a file system on page 44
- (Optional) Create an NMFS on page 45

Create a file system

File systems can be created only on nonroot metavolumes not in use. The metavolume must be at least 2 megabytes to accommodate a file system.

A file system name must be unique on a particular system. Name length is limited to 240 bytes (represented as 240 ASCII characters or a variable number of Unicode multibyte characters). A file system name can include upper and lowercase letters, numbers, hyphens (-), underscores (_), and periods (.). Alphanumeric names are accepted.

A file system name cannot:

- Begin with a hyphen, period, or the word root
- Include a blank character or colon (:)
- Be a single integer
- Be comprised entirely of integers

Action

To create a file system, use this command syntax:

```
$ nas_fs -name <fs_name> -create <volume_name>
```

where:

<fs_name> = name of the file system

<volume_name> = name of the existing volume

Example:

To create a file system called ufs1 by using existing volumes, type:

```
$ nas_fs -name ufs1 -create mtv1
```

Output

```
id          = 18
name        = ufs1
acl         = 0
in_use      = False
type        = uxfs
volume      = mtv1
rw_servers=
ro_servers=
rw_vdms     =
ro_vdms     =
symm_devs =
  002806000209-006,002806000209-007,002806000209-008,002806000209-009
disks       = d3,d4,d5,d6
```

(Optional) Create a mount point

Before you use a newly created file system, you can optionally create a mount point on a Data Mover. The server_mount command creates the mount point if a mount point does not exist.

A mount point name must begin with a slash (/), followed by alphanumeric characters (for example, /new). The name can include multiple components that make the mounted file system appear to be a logical subdirectory (for example, /new1/new2 or /new1/new2/new3/new4). A mount point name can have a maximum of seven components (for example, new1/new2/new3/.../new7).

With the exception of NMFS, mount a file system only on the last component of a multicomponent mount point name. No file system should be mounted on intermediate components of a mount point name. For example, if you have a file system mounted on the /new1 mount point, do not mount a file system on a mount point named /new1/new2.

Action

To create a mount point on a Data Mover, use this command syntax:

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

where:

<movername> = name of the Data Mover

<pathname> = path to mount point

Example:

To create a mount point named server_3, type:

```
$ server_mountpoint server_3 -create /ufs1
```

Output

```
server_3: done
```

Mount a file system

Before you begin

Before mounting a file system on a specific mount point, you need to know the name of:

- The Data Mover that contains the mount point
- The mount point on the Data Mover
- The file system that you want to mount

Note: The server_mount command creates a mount point if one does not exist.

Commands to obtain the information you need to mount a file system:

- nas_fs -list command to list existing file systems
- nas_server -list command to view Data Mover names
- server_mountpoint <movername> -list command to view mount points on a Data Mover

A file system can be mounted read/write on one Data Mover (default) or read-only on multiple Data Movers concurrently. When a file system is mounted read/write on a Data Mover, only that Data Mover is allowed access to the file system. No other Data Mover is allowed access.

When a file system is mounted read-only on a Data Mover, clients cannot write to the file system regardless of the client's access permissions. A file system can be mounted read-only on multiple Data Movers concurrently, as this is the default for checkpoint and TimeFinder/FS file systems.

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.

Procedure

If you create a mount point on a Data Mover, mount the file system on that mount point.

The -option argument is used to specify a number of mount options. The *VNX Command Line Interface Reference for File* provides a complete list of mount options available.

Action
To mount a file system on a mount point that is on a Data Mover, use this command syntax: `$ server_mount <movername> -option <options> <fs_name> <mount_point>` where: `<movername>` = name of the Data Mover `<options>` = list of mount options separated by comma `<fs_name>` = name of the file system

Action

<mount_point> = path to mount point for the Data Mover; a <mount_point> must begin with a forward slash (/)

Example:

To mount ufs1 on mount point /ufs1 with access checking policy set to NATIVE and nooplock turned on, type:

```
$ server_mount server_2 -option accesspolicy=NATIVE,nooplock ufs1 /ufs1
```

Output

```
server_2: done
```

(Optional) Create an NMFS

An NMFS must be mounted read-only on the Data Mover.

Use the server_mount command to mount the file system for read-only access on each Data Mover from which you want to provide read-only access to the file system. Unmount all file systems from a Data Mover on page 76 explains how to unmount a file system from a Data Mover:

1. Create an NMFS by using this command syntax:

   ```
   $ nas_fs -name <name> -type nmfs -create
   ```

   where:

   <name> = name of the NMFS

   Example:

   To create an NMFS named nmfs1, type:

   ```
   $ nas_fs -name nmfs1 -type nmfs -create
   ```

   Output:

   ```
   id         = 26
   name       = nmfs1
   acl        = 0
   in_use     = False
   type       = nmfs
   worm       = off
   volume     = 0
   pool       =
   rw_servers=
   ro_servers=
   rw_vdms    =
   stor_devs  =
   ro_vdms    =
   stor_devs  =
   disks      =
   ```

   Note: There is no volume allocation.

2. Create a mount point in the root of the designated Data Mover for the new file system.

3. Mount the NMFS as read-only on the Data Mover by using this command syntax:

   ```
   $ server_mount <movername> -option <options> <fs_name> <mount_point>
   ```

   where:

   <movername> = name of the Data Mover

   <options> = list of mount options separated by comma

   <fs_name> = name of the NMFS

   <mount_point> = path to mount point for the Data Mover; a *<mount_point>* must begin with a forward slash (/)

   Example:

   To mount an NMFS named nmfs1 as read-only on server_3, type:

   ```
   $ server_mount server_3 -option ro nmfs1 /nmfs1
   ```

   Output:

   ```
   server_3: done
   ```

4. Export the new file system for NFS access.

   or

   Share the file system for CIFS access.

Create a new component file system

The steps to create a new component file system and to mount it on an NMFS are similar to steps followed for mounting any file system:

1. Create a volume for the component file system.
2. Create the component file system on the new volume by using this command syntax:

   ```
   $ nas_fs -name <name> -create <volume_name>
   ```

   where:

   <name> = name assigned to the file system

   <volume_name> = name of the volume

   Example:

   To create a component file system called ufs1 on volume mtv1, type:

   ```
   $ nas_fs -name ufs1 -create mtv1
   ```

3. Mount the component file system to the NMFS by using this command syntax:

   ```
   $ server_mount <movername> -option <options> <fs_name> <mount_point>
   ```

   where:

<movername> = name of the Data Mover

<options> = list of mount options separated by comma

<fs_name> = name of the NMFS

<mount_point> = pathname of the NMFS which is in the format /nmfs path/component file system name

Example:

To mount ufs1, as a part of the NMFS nmfs1, type:

```
$ server_mount server_2 ufs1 /nmfs1/ufs1
```

Output:

```
server_2: done
```

5

Managing Volumes

This section provides information on how to manage existing volumes manually. Unless otherwise noted, the procedures apply to all the system models.

Note: List volumes on page 33 describes how to view a list of volumes.

The tasks to manage volumes are:

- Check volume capacity on page 50
- Rename a volume on page 50
- Clone a volume on page 51
- Delete a metavolume or stripe volume on page 52
- Delete a slice volume on page 54

Check volume capacity

Action

To check the volume capacity, use this command syntax:

```
$ nas_volume -size <volume_name>
```

where:

<volume_name> = name of the volume

Example:

To check the volume capacity of mtv1, type:

```
$ nas_volume -size mtv1
```

Output

```
size = total = 34524 avail = 34524 used = 0 ( 0% )
(sizes in MB)
```

Rename a volume

Action

To rename a volume, use this command syntax:

```
$ nas_volume -rename <old_name> <new_name>
```

where:

<old_name> = current name of the volume

<new_name> = new name of the volume

Example:

To rename the mtv metavolume to mtv1, type:

```
$ nas_volume -rename mtv mtv1
```

Output

```
id          = 247
name        = mtv1
acl         = 0
in_use      = False
type        = meta
volume_set  = stv1
disks       = d3,d4,d5,d6
```

Clone a volume

You can make an exact copy of a stripe volume, slice volume, or metavolume by cloning it. Cloning duplicates only the volume structure. It does not copy the file system or the data in the file system at the time of cloning.

If -option disktype and source_volume:destination_volume are used together, the behavior differs depending on which option is specified first.

Action

To clone a volume, use this command syntax:

```
$ nas_volume -Clone <volume_name> -option disktype=<disktype>
<source_volume>:<destination_volume>,...
```

where:

`<volume_name>` = name of the volume to be cloned

`<disktype>` = type of disk to be created

`<source_volume>` = sets a specific disk volume set for the source volume

`<destination_volume>` = sets a specific disk volume set for the destination volume

Example:

To clone the metavolume mtv1, type:

```
$ nas_volume -Clone mtv1
```

Output

```
id           = 127
name         = mtv1
acl          = 0
in_use       = False
type         = meta
volume_set   = d7
disks        = d7

id           = 128
name         = v128
acl          = 0
in_use       = False
type         = meta
volume_set   = d8
disks        = d8
```

Note

The example clones the metavolume mtv1. The default name of the cloned metavolume is v128.

Delete a metavolume or stripe volume

You cannot delete a metavolume or stripe volume in use by a file system. The tasks to delete a metavolume or stripe volume are:

- List metavolume information on page 52
- Delete a metavolume or stripe volume on page 53

List metavolume information

To ensure that the metavolume or stripe volume that you want to delete is not in use, list the volume information and check the in_use parameter.

Action

To list the volume information, use this command syntax:

```
$ nas_volume -info <volume_name>
```

where:

<volume_name> = name of the metavolume or stripe volume

Example:

To list the volume information for mtv1, type:

```
$ nas_volume -info mtv1
```

Output

```
id         = 247
name       = mtv1
acl        = 0
in_use     = False
type       = meta
volume_set = stv1
disks      = d3,d4,d5,d6
```

Note

The in_use parameter for the mtv1 metavolume is False, indicating that the metavolume is not in use by a file system.

Delete a metavolume or stripe volume

Remove all file systems from a volume that you want to delete. If the volume is part of a larger metavolume configuration, remove file systems from the larger metavolume and delete the larger metavolume before deleting the volume.

Action

To delete a metavolume, use this command syntax:

```
$ nas_volume -delete <volume_name>
```

where:

<volume_name> = name of the metavolume to delete

Example:

To delete a metavolume named mtv1, type:

```
$ nas_volume -delete mtv1
```

Output

```
id         = 146
name       = mtv1
acl        = 1432, owner=nasadmin, ID=201
in_use     = False
type       = meta
volume_set = d7,mtv1
disks      = d7,d8
```

Note

The in_use parameter for the mtv1 metavolume is False, indicating that the metavolume is not in use by a file system.

Delete a slice volume

You cannot delete a slice volume that is in use by a file system or metavolume. The tasks to delete a slice volume are:

- List slice volume information on page 54
- Delete a slice volume on page 55

List slice volume information

To ensure that the slice volume you want to delete is not in use, list the volume information and check the in_use parameter.

Action

To list the volume information, use this command syntax:

```
$ nas_slice -info <slice_name>
```

where:

<slice_name> = name of the slice volume

Example:

To list the slice volume information for slv1, type:

```
$ nas_slice -info slv1
```

Output

```
id          = 67
name        = slv1
acl         = 0
in_use      = False
type        = slice
slice_of    = d7
offset(MB)= 0
size (MB)   = 2048
volume_name = slv1
```

Note

The in_use parameter for the slv1 slice volume is False, indicating that the slice volume is not in use by a file system.

Delete a slice volume

If the slice volume is part of a metavolume configuration, remove file systems from the metavolume and delete the metavolume before deleting the slice volume.

Action

To delete a slice volume, use this command syntax:

```
$ nas_slice -delete <slice_name>
```

where:

<slice_name> = name of the slice volume to delete

Example:

To delete slice volume information for slv1, type:

```
$ nas_slice -delete slv1
```

Output

```
id          = 67
name        = slv1
acl         = 0
in_use      = False
slice_of    = d7
offset(MB)= 0
size (MB)  = 2048
```

Note

The in_use parameter for the slv1 slice volume is False, indicating that the slice volume is not in use by a file system.

6

Managing File Systems

This section provides information on how to manage existing file systems. Unless otherwise noted, these procedures apply to all system models.

The tasks to manage file systems are:

- Export a file system on page 58
- List file systems on page 60
- View the file system configuration information on page 61
- List mount points on page 61
- List mounted file systems on page 62
- Check disk space capacity for a single mounted file system on page 63
- Check disk space capacity for all mounted file systems on page 63
- Check inode capacity on page 64
- Extend a file system on page 66
- Extend replicated file systems on page 67
- Adjust file system size threshold on page 70
- Add an existing file system to an NMFS on page 71
- Move an NMFS on page 72
- Rename a file system on page 73
- Enhance file read/write performance on page 74
- Unmount all file systems from a Data Mover on page 76
- Delete a file system or NMFS on page 78

Export a file system

To make a file system available to NFS users, you must export a path to the file system from a Data Mover by using the server_export command. Each time the server_export command is run, an entry is added to the existing entries in an export table. Entries to the table are permanent and are automatically reexported when the system restarts.

Use the -option argument to specify:

- Level of access
- Authentication method for each exported file system
- Whether the exported file system can be accessed by using only NFSv4

For CIFS clients, the component file system is shared.

The tasks to export a file system are:

- Export a file system from a Data Mover for NFS access on page 58
- Export a file system from a Data Mover for CIFS access on page 59
- Export an NMFS on page 59

Export a file system from a Data Mover for NFS access

Action

To export a file system for NFS access, use this command syntax:

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

where:

<movername> = name of the Data Mover

<options> = NFS export options

<pathname> = pathname of the file system to mount

Example:

To export the file system ufs2 on an NFS client, type:

```
$ server_export server_3 -Protocol nfs -option root=10.1.1.1 /ufs2
```

Output

```
server_3: done
```

Export a file system from a Data Mover for CIFS access

Action

To export a file system for CIFS access, use this command syntax:

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

where:

<movername> = name of the Data Mover

<sharename> = name of the shared component file system

<pathname> = pathname of the file system to mount

Example:

To export the file system ufs2 on a CIFS client, type:

```
$ server_export server_3 -Protocol cifs -name ufs2 /ufs2
```

Output

```
server_3: done
```

Export an NMFS

When you export an NMFS, you export and mount the NMFS root which provides access to all component file systems. Any options set on the NMFS root propagate to the component file systems. However, you can export the component file system with different export options.

When you export a component file system in an NMFS hierarchy, you can export only the mount point path of the component file system. Subdirectories of the component file system cannot be exported.

List file systems

Action

To view a list of all file systems, type:

```
$ nas_fs -list
```

Output

```
id inuse type acl volume name             server
1  n     1    0   166    root_fs_1
2  y     1    0   168    root_fs_2        1
3  y     1    0   170    root_fs_3        2
4  y     1    0   172    root_fs_4        3
5  y     1    0   174    root_fs_5        4
6  n     1    0   176    root_fs_6
7  n     1    0   178    root_fs_7
8  n     1    0   180    root_fs_8
9  n     1    0   182    root_fs_9
10 n     1    0   184    root_fs_10
11 n     1    0   186    root_fs_11
12 n     1    0   188    root_fs_12
13 n     1    0   190    root_fs_13
14 n     1    0   192    root_fs_14
15 n     1    0   194    root_fs_15
16 y     1    0   196    root_fs_common   4,3,2,1
17 n     5    0   245    root_fs_ufslog
18 n     1    0   247    ufs1
```

Note

Column definitions:

id — ID of the file system (assigned automatically)

inuse — Indicates whether the file system registered into the mount table of a Data Mover; y indicates yes, n indicates no

type — Type of file system

acl — Access control value for the file system

volume — Volume on which the file system resides

name — Name assigned to the file system

server — ID of the Data Mover that is accessing the file system

View the file system configuration information

If a file system is created by using the AVM feature, it has an associated storage pool. The pool option identifies the storage pool associated with that file system. Otherwise, the pool option has no value or is blank.

Action

To view configuration information of a specific file system, use this command syntax:

```
$ nas_fs -info <fs_name>
```

where:

<fs_name> = name of the file system

Example:

To view configuration information on ufs1, type:

```
$ nas_fs -info ufs1
```

Output

```
id          = 18
name        = ufs1
acl         = 0
in_use      = False
type        = uxfs
volume      = mtv1
pool        =
rw_servers  =
ro_servers  =
rw_vdms     =
ro_vdms     =
symm_devs   =
002806000209-006,002806000209-007,002806000209-008,002806000209-009
disks       = d3,d4,d5,d6
```

List mount points

Action

To list the mount points on a Data Mover, use this command syntax:

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

where:

<movername> = name of the Data Mover

Example:

To list mount points on server_3, type:

```
$ server_mountpoint server_3 -list
```

Output

```
server_3:
/.etc_common
/ufs1
/ufs1_snap1
```

List mounted file systems

You can view a list of all file systems currently mounted on a specific Data Mover and the options assigned to each of them.

Action

To display a list of all file systems mounted on a Data Mover, use this command syntax:

```
$ server_mount <movername>
```

where:

<movername> = name of the Data Mover

Example:

To list all file systems mounted on server_3, type:

```
$ server_mount server_3
```

Output

```
server_3:
fs2 on /fs2 uxfs,perm,rw
fs1 on /fs1 uxfs,perm,rw
root_fs_3 on / uxfs,perm,rw
```

Check disk space capacity for a single mounted file system

You must maintain file system space with care because it is a limited resource. The nas_fs command lets you view the disk space on the file system, that is, the total amount of allocated space, the current free and used space, and the percentage of used space.

Action

To display disk space of a file system, use this command syntax:

```
$ nas_fs -size <fs_name>
```

where:

<fs_name> = name of the file system

Example:

To view the total space available on ufs1, type:

```
$ nas_fs -size ufs1
```

Output

```
total = 2041959 avail = 2041954 used = 6 ( 0% )(sizes in MB)
(blockcount = 4246732800 )
volume: total = 2073600 (sizes in MB) ( blockcount = 4246732800 )
root_fs_3 on / uxfs,perm,rw
```

Check disk space capacity for all mounted file systems

Action

To display the total disk space of all file systems on a Data Mover, use this command syntax:

```
$ server_df <movername>
```

where:

<movername> = name of the Data Mover

Example:

To view the total disk space of all file systems on server_2, type:

```
$ server_df server_2
```

Output

```
server_2:
Filesystem     kbytes     used  avail     capacity Mounted on
root_fs_common 15360      1392  13968     9%       /.etc_common
ufs1           34814592   54240 34760352  0%       /ufs1
ufs2           104438672  64    104438608 0%       /ufs2
ufs1_snap1     34814592   64    34814528  0%       /ufs1_snap1
root_fs_2      15360      224   15136     1%       /
```

Check inode capacity

The number of inodes defines the number of filenames and directory names that the file system can contain. You can check the inode capacity of a single file system or all file systems on a specific Data Mover. A specific number of inodes is allocated to a file system when you create it.

The nas_fs command lists the total number of inodes allocated to the file system, the available and used inodes, and the percentage of total inodes in use by the file system.

The tasks to check the inode capacity of a file system on a Data Mover are:

- View the inode capacity of a single file system on a Data Mover on page 64
- View the inode capacity of all file systems on a Data Mover on page 65

View the inode capacity of a single file system on a Data Mover

Action

To view the inode capacity on a single Data Mover, use this command syntax:

```
$ server_df <movername> -inode <fs_name>
```

where:

<movername> = name of the Data Mover

<fs_name> = name of the file system

Example:

To view the inode allocation and availability on ufs2, type:

```
$ server_df server_2 -inode ufs2
```

Output

```
server_2:
Filesystem     inodes   used avail    capacity Mounted on
ufs2           12744766 8    12744758 0%       /ufs2\
```

Note

Column definitions:

Filesystem — Name of the file system

inodes — Total number of inodes allocated to the file system

used — Number of inodes in use by the file system

avail — Number of free inodes available for use by the file system

capacity — Percentage of total inodes in use

Mounted on — Name of the file system mount point on the Data Mover

View the inode capacity of all file systems on a Data Mover

Action

To view the inode capacity of all file systems on a Data Mover, use this command syntax:

`$` **`server_df`** *`<movername>`* **`-inode`**

where:

`<movername>` = name of the Data Mover

Example:

To view the inode capacity of all file systems on server_2, type:

`$ server_df server_2 -inode`

Output

```
server_2:
Filesystem     inodes   used avail    capacity Mounted on
root_fs_common 7870     14   7856     0%       /.etc_common
ufs1           4250878  1368 4249510  0%       /ufs1
ufs2           12744766 8    12744758 0%       /ufs2
ufs1_snap1     4250878  8    4250870  0%       /ufs1_snap1
root_fs_2      7870     32   7838     0%       /
```

Extend a file system

Before you begin

When a file system nears its maximum capacity, you can increase the file system size by adding volumes. To increase the total capacity of the NMFS, extend an existing component file system or add new component file systems. You can also extend a file system with any unused disk volume, slice volume, stripe volume, or metavolume. Adding volume space to a file system adds the space to the metavolume on which it is built. So, when you extend a file system, the total size of its underlying metavolume is also extended.

When you extend a file system, extend it with the same volume type as the original file system. For example, if the metavolume underlying the file system is made up of stripe volumes, you should extend the file system with stripe volumes of the same size and type.

File systems created with AVM have associated storage pools. These file systems can be extended by size. View the file system configuration information on page 61 provides the information to determine whether a file system has an associated pool. *Managing Volumes and File Systems with VNX AVM* provides additional information on extending file systems created with AVM.

Procedure

1. Check the size of the file system before extending it by using this command syntax:

   ```
   $ nas_fs -size <fs_name>
   ```

   where:

   `<fs_name>` = name of the file system

   Example:

   To check the size of the file system ufs1, type:

   ```
   $ nas_fs -size ufs1
   ```

   Output:

   ```
   total = 67998 avail = 67997 used = 0 (0%) (sizes in MB)
   volume: total = 69048 (sizes in MB)
   ```

2. Extend the file system by using this command syntax:

   ```
   $ nas_fs -xtend <fs_name> <volume_name>
   ```

   where:

   `<fs_name>` = name of the file system

   `<volume_name>` = name of the volume

   Example:

   To extend the file system ufs1, type:

   ```
   $ nas_fs -xtend ufs1 emtv2b
   ```

Output:

```
id        = 18
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
volume    = mtv1, emtv2b
profile   =
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
symm_devs = 002804000190-0034,002804000190-0035,002804000190-0036,
002804000190-0037,002804000190-0040,002804000190-0041,002804000190-0042,
002804000190-0043
disks     = d3,d4,d5,d6,d15,d16,d17,d18
 disk=d3  symm_dev=002804000190-0034  addr=c0t318-15-0  server=server_2
 disk=d4  symm_dev=002804000190-0035  addr=c0t319-15-0  server=server_2
 disk=d5  symm_dev=002804000190-0036  addr=c0t3l10-15-0 server=server_2
 disk=d6  symm_dev=002804000190-0037  addr=c0t3l11-15-0 server=server_2
 disk=d15 symm_dev=002804000190-0040  addr=c0t4l4-15-0  server=server_2
 disk=d16 symm_dev=002804000190-0041  addr=c0t4l5-15-0  server=server_2
 disk=d17 symm_dev=002804000190-0042  addr=c0t4l6-15-0  server=server_2
 disk=d18 symm_dev=002804000190-0043  addr=c0t4l7-15-0  server=server_2
```

3. Check the size of the file system after extending it by using this command syntax:

   ```
   $ nas_fs -size <fs_name>
   ```

   where:

   <fs_name> = name of the file system

   Example:

   To check the size of the file system ufs1 after extending it, type:

   ```
   $ nas_fs -size ufs1
   ```

   Output:

   ```
   total = 138096 avail = 138096 used = 0 ( 0% ) (sizes in MB)
   volume: total = 138096 (sizes in MB)
   ```

Extend replicated file systems

Replicated file systems can be manually extended when VNX Replicator is running. A slice volume is required to extend a source file system.

Before you extend a source file system:

- Use the nas_fs -size command to verify the current size of the file system. When you extend the source file system by using the -xtend option of the nas_fs command, the destination file system is extended first and then the source file system. This operation maintains identical file system sizes at the source and destination sites.

- You can extend the size of the source (production) file system without impacting the destination file system by using the -xtend src_only option. The *VNX Command Line Interface Reference for File* provides a detailed synopsis of the commands associated with the VNX Replicator.
- Verify whether there is enough volume space to extend the source and destination file systems.

1. On the primary site, verify the current sizes of the source and destination file systems by using this command syntax:

   ```
   $ nas_fs -size <fs_name>
   ```

   where:

   <fs_name> = name of the file system

   Example:

   To verify the current size of the source file system src_ufs1, type:

   ```
   $ nas_fs -size src_ufs1
   ```

   Output:

   ```
   total = 67998 avail = 67997 used = 0 (0%) (sizes in MB)
   volume: total = 69048 (sizes in MB)
   ```

2. On the remote site, verify the current sizes of the source and destination file systems by using this command syntax:

   ```
   $ server_df <fs_name>
   ```

   where:

   <fs_name> = name of the file system

   Example:

   To verify the current size of the source and destination file system dst_ufs1, type:

   ```
   $ server_df server_2 dst_ufs1
   ```

   Output:

   ```
   Filesystem    kbytes    used      avail    capacity  Mounted on
   ppfs1         4130288   97392    4032896     2%        /pfs1
   ```

3. Extend the source file system (on the primary site) by using this command syntax:

   ```
   $ nas_fs -xtend <fs_name> <volume_name>
   ```

   where:

   <fs_name> = name of the source file system

   <volume_name> = name of the volume

   Example:

   To extend the source file system (on the primary site), type:

```
$ nas_fs -xtend src_ufs1 mtv4
```

4. Check the size of the file system after extending it by using this command syntax:

   ```
   $ nas_fs -size <fs_name>
   ```

   where:

   <fs_name> = name of the source file system

   Example:

   To check the size of the source file system src_ufs1 after extending it, type:

   ```
   $ nas_fs -size src_ufs1
   ```

   Output:

   ```
   total = 138096 avail = 138096 used = 0 ( 0% ) (sizes in MB)
   volume: total = 138096 (sizes in MB)
   ```

Adjust file system size threshold

Because file system performance can degrade as its used space approaches 100 percent full, the VNX system monitors the amount of space in use in its file systems and triggers an event when the used space in a file system exceeds 90 percent. To get a notification of this event, you need to set up event logging (an SNMP trap or email notification) as described in *Configuring Events and Notifications on VNX for File*.

When the file system size threshold is exceeded, there are corrective actions you can take to manage the issue. You can either move files from the file system or extend the file system size. Either action helps reduce the percentage of used space to a percentage below the threshold.

For example, if you anticipate that the amount of used space in your file system might grow at a rapid rate, you can decrease the file system size threshold to a smaller percentage.

The tasks to adjust file system size threshold are:

- Adjust the file system size threshold for all file systems on page 70
- Adjust the file system size threshold on a single Data Mover on page 71

Adjust the file system size threshold for all file systems

1. Change the size threshold for all file systems by using this command syntax:

```
$ server_param ALL -facility <facility_name> -modify <param_name>
-value <new_value>
```

where:

`<movername>` = name of the Data Mover

`<facility_name>` = name of the facility to which the parameter belongs

`<param_name>` = name of the parameter

`<new_value>` = new value for the parameter

Example:

To change the size threshold for all file systems to 85 percent, type:

```
$ server_param ALL -facility file -modify fsSizeThreshold -value 85
```

2. Restart the Data Movers by typing:

```
$ server_cpu ALL -reboot now
```

Adjust the file system size threshold on a single Data Mover

1. Change the file system size threshold on a single Data Mover by using this command syntax:

   ```
   $ server_param <movername> -facility <facility_name> -modify <param_name>
   -value <new_value>
   ```

   where:

   `<movername>` = name of the Data Mover

   `<facility_name>` = name of the facility to which the parameter belongs

   `<param_name>` = name of the parameter

   `<new_value>` = new value for the parameter

   Example:

   To change the size threshold for all file systems on server_2, type:

   ```
   $ server_param server_2 -facility file -modify fsSizeThreshold -value 85
   ```

2. Restart the Data Mover for the change to take effect by using this command syntax:

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

   where:

   `<movername>` = name of the Data Mover

   Example:

   To restart server_2 for the change to take effect, type:

   ```
   $ server_cpu server_2 -reboot now
   ```

Add an existing file system to an NMFS

You can add an existing file system to an NMFS or remove a file system from an NMFS without modifying or changing the file system:

1. Permanently unmount the file system from a Data Mover by using this command syntax:

   ```
   $ server_umount <movername> -perm <fs_name>
   ```

   where:

   `<movername>` = name of the Data Mover

   `<fs_name>` = name of the file system to unmount

   Note: To permanently unmount a file system from a Data Mover by specifying the mount point path, use the -perm <mount_point> option instead of the -perm <fs_name> option.

Example:

To permanently unmount a file system named fs1, type:

```
$ server_umount server_2 -perm /fs1
```

Output:

```
server_2: done
```

2. Create a new mount point for the file system in the NMFS.

3. Mount the file system in the NMFS by using this command syntax:

```
$ server_mount <movername> -option <options> <fs_name> <mount_point>
```

where:

`<movername>` = name of the Data Mover

`<options>` = list of mount options separated by comma

`<fs_name>` = name of the NMFS

`<mount_point>` = pathname of the NMFS which is in the format /nmfs path/component file system name

Example:

To mount a file system on a mount point on server_3 with a nolock option, type:

```
$ server_mount server_3 -option nolock fs5/nmfs4/fs5
```

Output:

```
server_2: done
```

Move an NMFS

You can move an NMFS from one Data Mover to another:

1. Permanently unmount each of the component file systems.
2. Permanently unmount the NMFS.
3. Mount the NMFS on the new Data Mover.
4. Mount each component file system on the NMFS on the new Data Mover.

Rename a file system

Action

To rename a file system, use this command syntax:

```
$ nas_fs -rename <old_name> <new_name>
```

where:

<old_name> = existing name of the file system

<new_name> = new name of the file system

Example:

To rename a file system ufs as ufs1, type:

```
$ nas_fs -rename ufs ufs1
```

Output

```
id = 18
name = ufs1
acl = 0
in_use = False
type = uxfs
volume = mtv1
rw_servers=
ro_servers=
rw_vdms    =
ro_vdms    =
symm_devs =
002806000209-006,002806000209-007,002806000209-008,002806000209-009
disks = d3,d4,d5,d
```

Enhance file read/write performance

The system includes internal mechanisms for enhancing read/write performance characteristics for certain types of files. The read prefetch mechanism is designed to optimize read operations of large files. It can speed up reading to 100 percent on large files. This mechanism is turned on by default on the server. It should be turned off only if the read access pattern for files in the file system primarily consists of small random accesses. The tasks to enhance file read/write performance are:

- Turn off read prefetch for a specific file system on page 74
- Turn off read prefetch for all file systems on a Data Mover on page 75
- Enable uncached write mechanism on page 76

Turn off read prefetch for a specific file system

Action

To turn off the read prefetch mechanism, use this command syntax:

```
$ server_mount <movername> -option <options>,noprefetch <fs_name> <mount_point>
```

where:

<movername> = name of the Data Mover

<options> = specifies mount options, separated by commas

<fs_name> = name of the file system

<mount_point> = path to mount point for the Data Mover; a <mount_point> must begin with a forward slash (/)

Example:

To turn off the read prefetch mechanism for ufs1, type:

```
$ server_mount server_3 -option rw,noprefetch ufs1 /ufs1
```

Output

```
server_3: done
```

Turn off read prefetch for all file systems on a Data Mover

1. Turn off the read prefetch mechanism for all file systems on a Data Mover by using this command syntax:

   ```
   $ server_param <movername> -facility <facility_name> -modify
   prefetch -value 0
   ```

   where:

   <movername> = name of the Data Mover

   <facility_name> = facility for the parameters

   Example:

   To turn off the prefetch mechanism for all file systems on server_2, type:

   ```
   $ server_param server_2 -facility file -modify prefetch -value 0
   ```

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

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

   where:

   <movername> = name of the Data Mover

   Example:

   To restart server_2 immediately, type:

   ```
   $ server_cpu server_2 -reboot now
   ```

Enable uncached write mechanism

The write mechanisms are designed to improve performance for applications, such as databases, with many connections to a large file. These mechanisms can enhance database access through the NFS protocol by 30 percent or more. The mechanism is turned off by default. However, it can be turned on for a file system.

Action

To turn on the uncached write mechanism for a file system, use this command syntax:

```
$ server_mount <movername> -option <options>,uncached <fs_name> <mount_point>
```

where:

`<movername>` = name of the Data Mover

`<options>` = specifies mount options, separated by commas

`<fs_name>` = name of the file system

`<mount_point>` = path to mount point for the Data Mover; a <mount_point> must begin with a forward slash (/)

Example:

To turn on the uncached write mechanism for the file system ufs1, type:

```
$ server_mount server_3 -option rw,uncached ufs1 /ufs1
```

Output

```
server_3: done
```

Unmount all file systems from a Data Mover

Unexport NFS exports and unshare CIFS shares of the file systems before you unmount all file systems on the Data Mover, particularly when you are unmounting file systems permanently.

To change the way in which a file system is mounted, use the server_umount command to unmount the file system permanently from the Data Mover and then remount the file system.

You can also unmount component file systems from the NMFS.

The tasks to unmount all file systems are:

- Unmount all file systems temporarily on page 77
- Unmount all file systems permanently on page 77

Unmount all file systems temporarily

The -temp option of the server_umount command is the default and does not need to be specified as part of the command.

Action

To temporarily unmount all file systems on a Data Mover, use this command syntax:

```
$ server_umount <movername> -temp -all
```

where:

`<movername>` = name of the Data Mover

Example:

To temporarily unmount all file systems on server_2, type:

```
$ server_umount server_2 -temp -all
```

Output

```
server_2: done
```

Unmount all file systems permanently

Permanently unmounting all file systems from a Data Mover deletes the contents of the mount table. To reestablish client access to the file systems, you must rebuild the mount table by remounting each file system on the Data Mover.

Action

To permanently unmount all file systems on a Data Mover, use this command syntax:

```
$ server_umount <movername> -perm -all
```

where:

`<movername>` = name of the Data Mover

Action

Example:

To permanently unmount all file systems on server_2, type:

```
$ server_umount server_2 -perm -all
```

Output

```
server_2: done
```

Delete a file system or NMFS

To delete a file system and free its disk space, you must delete or disconnect all entities associated with the file system: all checkpoints, BCVs, slice volumes, stripe volumes, and metavolumes. After you delete or disconnect all the file system entities, the disk volumes that provide storage space to the file system become part of the available free space on the file server.

Deleting a file system deletes all data on the file system. If there are checkpoints associated with the file system, delete the checkpoints before deleting the file system. Deleting the file system does not delete data from BCVs associated with the file system. *Using TimeFinder/FS, NearCopy, and FarCopy on VNX* provides information on how to unmirror a BCV.

To delete an NMFS, first permanently unmount all component file systems mounted in the NMFS and then permanently unmount the NMFS. Use the nas_fs command -delete option to delete the file system. Unmount all file systems from a Data Mover on page 76 provides additional information on unmounting file systems permanently:

1. Back up all the data you want to keep.
2. Check the file system configuration to determine if the file system has an associated storage pool (you will need this information in a later step) by using this command syntax:

   ```
   $ nas_fs -info <fs_name>
   ```

   where:

   <fs_name> = name of the file system

   If the pool output line displays a value, the file system has an associated storage pool.

   If the file system does not have an associated storage pool, proceed to step 3. If the file system has an associated storage pool, proceed to step 4.

3. Determine and notate the metavolume name on which the file system is built. You need to provide the metavolume name in step 10:

   ```
   $ nas_fs -info <fs_name>
   ```

   where:

   <fs_name> = name of the file system

   Note: The Volume field contains the metavolume name. The Disks field lists the disks providing storage to the file system.

4. If the file system has associated checkpoints, permanently unmount and then delete the checkpoints and their associated volumes.
5. If the file system has associated BCVs, break the connection between (unmirror) the file system and its BCVs.

Using TimeFinder/FS, NearCopy, and FarCopy on VNX provides information on how to unmirror a BCV.

6. If the file system is an NFS-exported file system, permanently disable client access to the file system by using this command syntax:

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

   where:

   `<movername>` = name of the Data Mover

   `<pathname>` = NFS entry

7. If the file system is a CIFS-exported file system, permanently disable client access to the file system by using this command syntax:

   ```
   $ server_export <movername> -Protocol cifs -unexport <sharename>
   ```

   where:

   `<movername>` = name of the Data Mover

   `<sharename>` = name of the shared component file system

8. Permanently unmount the file system from its associated Data Movers by using this command syntax:

   ```
   $ server_umount <movername> -perm <fs_name>
   ```

   where:

   `<movername>` = name of the Data Mover

   `<fs_name>` = name of the file system

   Note: To delete an NMFS, permanently unmount all component file systems in the NMFS.

9. Delete the file system or NMFS from the VNX system by using this command syntax:

   ```
   $ nas_fs -delete <fs_name>
   ```

   where:

   `<fs_name>` = name of the file system

   If the file system has an associated storage pool, as part of the file system delete operation, AVM deletes all underlying volumes and frees the space for use by other file systems.

   If the file system has no associated storage pool, proceed to step 10. The volumes underlying the file system were created manually and must be manually deleted.

10. Delete the metavolume on which the file system was created by using this command syntax:

    ```
    $ nas_volume -delete <volume_name>
    ```

where:

<volume_name> = name of the volume

11. If the metavolume included stripe volumes, delete all stripe volumes associated with the metavolume by using this command syntax, until the disk space is free:

```
$ nas_volume -delete <volume_name>
```

where:

<volume_name> = name of the volume

12. If the metavolume included slice volumes, delete all slice volumes associated with the metavolume by using this command syntax, until the disk space is free:

```
$ nas_volume -delete <volume_name>
```

where:

<volume_name> = name of the volume

13. After freeing disk space, check for slice volumes, stripe volumes, and metavolumes not in use (identified by an “n” in the **inuse** column in the command output) by using these commands:

```
$ nas_volume -list
```

```
$ nas_slice -list
```

Delete unused volumes until you free all the disk space you want.

7

Monitoring and Repairing File Systems

The tasks to check file system consistency and to repair a damaged file system are:

- Run a file system check on page 82
- Start an ACL check on the file system on page 82
- List file system checks on page 83
- Display the file system check information on a file system on page 83
- Display information on all current file system checks on page 84

Run a file system check

Action

To start and monitor fsck on a file system, use this command syntax:

```
$ nas_fsck -start <fs_name> -monitor
```

where:

<fs_name> = name of the file system

Example:

To start fsck on ufs1 and monitor the progress, type:

```
$ nas_fsck -start ufs1 -monitor
```

Output

```
id                             = 27
name                           = ufs1
volume                         = mtv1
fsck_server                    = server_2
inode_check_percent            = 10..20..30..40..60..70..80..100
directory_check_percent        = 0..0..100
used_ACL_check_percent         = 100
free_ACL_check_status          = Done
cylinder_group_check_status = In Progress..Done
```

Start an ACL check on the file system

The nas_fsck command allows you to manually start fsck on a specified file system. The nas_fsck command also lists and displays the status of fsck and aclchk. The aclchk utility finds and corrects any errors in the ACL database and removes duplicate ACL records stored on the file system. The aclchkonly option can only be used on a file system that is mounted but not exported. It cannot be run on an unmounted file system. By default, the fsck and aclchk utilities are enabled.

Action

To start an ACL check on a specified file system, use this command syntax:

```
$ nas_fsck -start <fs_name> -aclchkonly
```

where:

<fs_name> = name of the file system

Example:

To start an ACL check on ufs1 and monitor the progress, type:

```
$ nas_fsck -start ufs1 -aclchkonly
```

Output

```
ACLCHK: in progress for file system ufs1
```

List file system checks

Action

To list current file system checks, use this command syntax:

```
$ nas_fsck -list
```

Example:

To list current file system checks, type:

```
$ nas_fsck -list
```

Output

```
id      type  state         volume name             server
23      1     FSCK          134    ufs2             4
27      1     ACLCHK        144    ufs1             1
```

Display the file system check information on a file system

Action

To display file system check information on a single file system, use this command syntax:

```
$ nas_fsck -info <fs_name>
```

where:

<fs_name> = name of the file system

Example:

To display information about file system check for ufs2, type:

```
$ nas_fsck -info ufs2
```

Output

```
name                         = ufs2
id                           = 23
volume                       = v134
fsck_server                  = server_5
inode_check_percent          = 100
directory_check_percent      = 100
used_ACL_check_percent       = 100
free_ACL_check_status        = Done
cylinder_group_check_status  = In Progress
```

Display information on all current file system checks

Action

To display information on all file system checks that are currently running, use this command syntax:

```
$ nas_fsck -info -all
```

Example:

To display information on all file system checks that are currently running, type:

```
$ nas_fsck -info -all
```

Output

```
name                         = ufs2
id                           = 23
volume                       = v134
fsck_server                  = server_5
inode_check_percent          = 30
directory_check_percent      = 0
used_ACL_check_percent       = 0
free_ACL_check_status        = Not Started
cylinder_group_check_status = Not Started

name                         = ufs1
id                           = 27
volume                       = mtv1
fsck_server                  = server_2
inode_check_percent          = 100
directory_check_percent      = 0
used_ACL_check_percent       = 0
free_ACL_check_status        = Not Started
cylinder_group_check_status = Not Started
```

8

Troubleshooting

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

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

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

Topics included in this chapter are:

- EMC E-Lab Interoperability Navigator on page 86
- VNX user customized documentation on page 86
- Known problems and limitations on page 86
- Error messages on page 87
- EMC Training and Professional Services on page 87

EMC E-Lab Interoperability Navigator

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

VNX user customized documentation

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

Known problems and limitations

Table 6 on page 86 describes known problems that might occur when using volumes and file systems, and presents workarounds.

Table 6. Problems and workarounds

Known problem	Symptom	Workaround
The Unisphere software does not provide an interface for renaming a file system.		To rename a file system, you can type the appropriate CLI command in the CLI command entry page available in the Unisphere software, or directly in the CLI.
You are unable to mount a file system.	There are many probable causes for this scenario. An error message is displayed in most of the instances, though occasionally, there might not be one. In this case, the mount table entry already exists.	Perform server_mount -all to activate all entries in the mount table. Obtain a list of mounted file systems, and then observe the entries. If the file system in question is already mounted (temporary or permanent), perform the necessary steps to unmount it, and then retry.
An unmounted file system reappears in the mount table after the system restarts.	The file system might have been temporarily unmounted before the system restarts.	Perform a permanent unmount to remove the entry from the mount table.
When you create a new file in the NMFS root directory, a file exists error appears.	NMFS root directory is read-only.	Do not try to create files or folders in the NMFS root directory.

Table 6. Problems and workarounds *(continued)*

Known problem	Symptom	Workaround
You are unable to slice a disk volume.	You receive an error message and the slice will not be created.	To verify that the disk volume that you want to slice has enough unused space, use this command syntax: `$ nas_volume -size <volume_name>`

Error messages

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

To view message details, use any of these methods:

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

EMC Training and Professional Services

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

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

Appendix A

GID Support

Topics included are:

- Overview on page 90
- Restrictions for GID support on page 90

Overview

The system software supports 32-bit GID (group IDs) on NFS and CIFS file systems. This support enables a maximum GID value of 2,147,483,647 (approximately 2 billion).

Restrictions for GID support

With GID support, these restrictions apply:

- Enabling 16-bit GID support on a Data Mover does not decrease its maximum of 64,000 GIDs per file system. Regardless of the GID support setting (32-bit or 16-bit), there is a maximum limit of 64,000 GIDs per file system.
- File systems with 16-bit and 32-bit GIDs can coexist on a single Data Mover. Changing the gid32 parameter setting from 1 to 0 allows you to create file systems with 16-bit GIDs without disabling 32-bit GIDs on file systems already created with the parameter set to 1. Conversely, changing the gid32 parameter value from 0 to 1 allows you to create file systems with 32-bit GIDs without disabling 16-bit GID support on existing file systems.
- You cannot convert file systems created with 32-bit GID support to use 16-bit GIDs. Nor can you convert file systems created with 16-bit GID support to use 32-bit GIDs. The 32-bit GID support works only for file systems created with the parameter set to 1, and the 16-bit GID support works only for file systems created with the parameter set to 0.
- When backing up a file system with 32-bit GIDs, you risk truncating the GID values when the data is restored, if you use any of these server_archive formats:
  - emctar up to 31-bit
  - ustar up to 21-bit
  - cpio up to 15-bit
  - bcpio up to 16-bit
  - sv4cpio up to 21-bit
  - sv4crc up to 21-bit
  - tar up to 18-bit

  If you use these server_archive formats to back up file systems with 32-bit GIDs, a message appears to indicate that the UIDs are being forced to 0 and the GIDs are being forced to 1.
- Some backup applications have restrictions. Ensure that the application can handle 32-bit UIDs/GIDs.

There is no command to verify whether a file system supports 16-bit or 32-bit GIDs.

Glossary

A

append-only state

State of a file when the data in it cannot be modified, but the file can have new data appended to the end of it. In addition, the file itself cannot be deleted. Once a file in the append-only state has been written to, changing it to the locked state by making it read-only locks it into that state until its retention date has passed.

Automatic Volume Management (AVM)

Feature of VNX for File that creates and manages volumes automatically without manual volume management by an administrator. AVM organizes volumes into storage pools that can be allocated to file systems.

See also thin provisioning.

B

business continuance volume (BCV)

Symmetrix volume used as a mirror that attaches to and fully synchronizes with a production (source) volume on the VNX for File. The synchronized BCV is then separated from the source volume and is addressed directly from the host to serve in backup and restore operations, decision support, and application testing.

C

component file system

File system mounted on the nested mount root file system that is part of the nested mount file system.

D

disk volume

On a VNX for File, a physical storage unit as exported from the storage system. All other volume types are created from disk volumes.

See also *metavolume*, *slice volume*, *stripe volume*, and *volume*.

E

expired state

State of a file when its retention date has passed. A file in the expired state can be reverted back to the locked state or deleted from the FLR-enabled file system, but cannot be altered.

F

file system

Method of cataloging and managing the files and directories on a system.

file-level retention (FLR)

FLR lets you store data on standard rewritable magnetic disks by using NFS or CIFS operations to create a permanent, unalterable set of files and directories.

See also *append-only state*, *expired state*, *locked state*, *not locked state*, and *retention date*.

I

inode

"On-disk" data structure that holds information about files in a file system. This information identifies the file type as being a file that includes VNX FileMover stub files, a directory, or a symbolic link.

L

locked state

State of a file when its read/write permission is changed to read-only in a file system enabled for file-level retention. Files committed to the locked (WORM) state cannot be altered or deleted until their retention date has passed.

logical unit number (LUN)

Identifying number of a SCSI or iSCSI object that processes SCSI commands. The LUN is the last part of the SCSI address for a SCSI object. The LUN is an ID for the logical unit, but the term is often used to refer to the logical unit itself.

M

metavolume

On a VNX for File, a concatenation of volumes, which can consist of disk, slice, or stripe volumes. Also called a hypervolume or hyper. Every file system must be created on top of a unique metavolume.

See also *disk volume*, *slice volume*, *stripe volume*, and *volume*.

N

nested mount file system (NMFS)

File system that contains the nested mount root file system and component file systems.

nested mount file system root
File system on which the component file systems are mounted read-only, except for mount points of the component file systems.

not locked state
Initial state of a file when it is created. A file that is not locked is treated in the same manner as any file in a file system not enabled for file-level retention (FLR). This means it can be renamed, modified, and deleted until the time that the file is locked.

P

Production File System (PFS)
Production File System on VNX for File. A PFS is built on Symmetrix volumes or VNX for Block LUNs and mounted on a Data Mover in the VNX for File.

R

retention date
Date until which a locked file in an FLR-enabled file system will be protected. Users and applications manage a file's retention date by using NFS or CIFS to set the file's last access time to a future date and time. The retention timestamp is compared to the file system's FLR clock to determine whether a file's retention date has passed.

S

slice volume
On VNX for File, a logical piece or specified area of a volume used to create smaller, more manageable units of storage.

See also *disk volume*, *metavolume*, *stripe volume*, and *volume*.

storage pool
Groups of available disk volumes organized by AVM that are used to allocate available storage to file systems. They can be created automatically by AVM or manually by the user.

See also *Automatic volume management (AVM)*

stripe volume
Arrangement of volumes that appear as a single volume. Allows for stripe units that cut across the volume and are addressed in an interlaced manner. Stripe volumes make load balancing possible.

See also *disk volume*, *metavolume*, and *slice volume*.

V

volume
On VNX for File, a virtual disk into which a file system, database management system, or other application places data. A volume can be a single disk partition or multiple partitions on one or more physical drives.

See also *disk volume*, *metavolume*, *slice volume*, and *stripe volume*.

Index

A

adjusting file system size threshold 70, 71
automatic file system extension 30

B

business continuance volume (BCV)
    configuration 23

C

capacity
    volume 50
cautions 11, 12
    file systems 11
    fsck 12
    fsck processes 12
    nas_db database 11
    slice volumes 11
    spanning storage systems 11
CAVA
    integration considerations 11
cloning a volume 51
component file system
    creating 46
    extending 66
    unmounting 76
creating
    stripe volume 34, 35
    volumes 32

D

    listing
        mount points 61
        mounted file systems 62
deleting
    file system 78
    metavolume 52
    stripe volume 52
disk volume
    explanation 19
    freeing file system space 78
    renaming 50
    unused 32

E

EMC E-Lab Navigator 86
error messages 87
export NMFS 59

F

file system
    concepts 16
    deleting 78
    displaying mounted 62
    freeing disk space allocated 78
    mirroring 24
    permanent
        mount 44
    quotas 11, 29
    size guidelines 26
    unmount all
        permanent 77
        temporary 77
file system size threshold 70, 71
    change
        for Data Mover 70, 71
    fsSizeThreshold parameter 70, 71
file-level retention
    integration considerations 29
fsck

fsck *(continued)*
caution 12
repairing file systems 17
running file system check 17
starting an ACL check 18
system performance 12
fsSizeThreshold parameter 70, 71

G

GID support
restrictions 90

I

integration considerations
file-level retention 29
quotas 29
Replicator 30
SnapSure 30
SRDF 30
TimeFinder/FS 29
International character supportUnicode characters 11

M

messages, error 87
metavolume 16, 20, 21, 22, 35, 50, 52
addressing 22
concepts 16
configuration guidelines 21
creating 35
deleting 52
renaming 50
mirrored file system 24
mount point
creating 43
listing 61

N

nas_db database
cautions and restrictions 11
nas_fsck 82, 83, 84
nested mount file system (NMFS)
about 26
NMFS, export 59

P

parameters
fsSizeThreshold 70, 71
protocol
Trivial File Transfer Protocol (TFTP) 25

Q

quotas
integration considerations 11, 29
quotas for file system 11

R

RAID group combinations 38
renaming
disk volume 50
metavolume 50
slice volume 50
stripe volume 50
Replicator
integration considerations 30
restrictions
GID support 90
nas_db database 11
TimeFinder/FS 11

S

server_mount command 43, 74, 76
slice volume
how it works 19
renaming 50
SnapSure
integration considerations 11, 30
SRDF
integration considerations 30
stripe volume
creating 34, 35
deleting 52
explanation 20
improving performance with 20
renaming 50

T

TFTP. See Trivial File Transfer Protocol (TFTP) 25
TimeFinder/FS
integration considerations 11, 29
Trivial File Transfer Protocol (TFTP)
protocol 25
troubleshooting 85

U

unmounting file systems from Data Mover

unmounting file systems from Data Mover *(continued)*
- permanent 77
- temporary 77

V

VNX Replicator 30

volume
- available space 50
- BCV configuration 23
- capacity, checking 50
- cloning 51
- concepts 16
- creation 32
- deleting 52
- management, creating
  - metavolume 35

volume *(continued)*
- management, creating *(continued)*
  - stripe volume 34, 35
- metavolume
  - addressing 22
- metavolume configuration 21
- renaming 50
- size 50
- stripe volume
  - configuration guidelines 28
  - performance 20

volumes
- BCV 23
- cloning 51
- disk 19
- metavolume 21
- slice 19
- stripe 20