

Contents

Introduction to managing Celerra volumes and file systems manually . . .	3
Terminology	3
Restrictions	5
Cautions	5
Volumes and file system concepts	7
File systems	7
Volumes	9
System requirements for managing volumes and file systems manually	15
E-Lab Interoperability Navigator	15
Planning considerations for managing volumes and file systems manually	16
Supported file system access protocols	16
File system size guidelines	17
NMFS	17
Volume configuration guidelines	18
Stripe volume configuration considerations	19
Integration considerations	19
User interface choices for managing volumes and file systems manually	22
Roadmap for managing volumes and file systems manually	23
Creating volumes	24
Determine storage availability	24
Creating volumes	25
Adding disk volumes to a Celerra gateway system with a CLARiiON array	27
Creating file systems	30
Before creating a file system	30
Create a file system	31
(Optional) Creating a mount point	31
Mount a file system	32
(Optional) Create an NMFS	34

Managing file systems	36
Export a file system	36
List file systems	38
View the file system configuration information	39
List mount points	39
List mounted file systems	40
Check disk space capacity for a single mounted file system	40
Check disk space capacity for all mounted file systems	41
Check inode capacity	41
Extend a file system	43
Extend replicated file systems	46
Adjust file system size threshold	48
Add an existing file system to the NMFS	49
Move an NMFS	51
Rename a file system	51
Enhance file read/write performance	52
Unmount all file systems from a Data Mover	54
Delete a file system	55
Managing volumes	58
List volumes	58
Check volume capacity	59
Rename a volume	59
Clone a volume	60
Delete a metavolume or stripe volume	61
Delete a slice volume	62
Monitoring and repairing file systems	64
Before monitoring and repairing a file system	64
Run file system check	65
Start an ACL check on the file system	66
List file system checks	67
Display the file system check information on a file system	67
Display information on all current file system checks	68
Troubleshooting volumes and file systems	69
Where to get help	69
Known problems and limitations	70
Error messages for volumes and file systems	71
Related information	72
Customer training programs	72
Appendix A: GID support	73
Restrictions for GID support	73
Index	75

Introduction to managing Celerra volumes and file systems manually

The EMC® Celerra® Network Server allows you to create and manage Celerra volumes and file systems manually or automatically.

This document is part of the Celerra Network Server documentation set and is intended for system administrators responsible for creating and managing Celerra volumes and file systems manually.

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

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

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

Terminology

This section defines terms important to understanding how to manage volumes and file systems on the Celerra Network Server. The *EMC Celerra Glossary* provides a complete list of Celerra terminology.

Automatic Volume Management (AVM): Feature of the Celerra Network Server 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.

business continuance volume (BCV): Symmetrix volume used as a mirror that attaches to and fully synchronizes with a production (source) volume on the Celerra Network Server. 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.

CLEAN state: Initial state of a file when it is created. A CLEAN file 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 it is committed to FLR.

component file system: File system mounted on the nested mount root file system and is part of the nested mount file system.

disk volume: On Celerra systems, a physical storage unit as exported from the storage array. All other volume types are created from disk volumes. *See also* metavolume, slice volume, stripe volume, and volume.

EXPIRED state: State of a file when its retention period expires. A file in the EXPIRED state can be reverted to the FLR state or deleted from the FLR file system, but cannot be altered.

file-level retention (FLR): FLR lets you archive data on standard rewritable magnetic disks using NFS or CIFS operations to create a permanent, unalterable set of files and directories. *See also* CLEAN state, EXPIRED state, FLR state, and retention period.

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

FLR 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 FLR state cannot be altered or deleted until their retention period expires.

inode: “On-disk” data structure that holds information about files in a file system. This information identifies the file type as being a file including Celerra FileMover stub files, a directory, or a symbolic link.

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.

metavolume: On a Celerra system, 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.

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.

production file system (PFS): Production file system on a Celerra Network Server. A PFS is built on Symmetrix volumes or CLARiiON LUNS and mounted on a Data Mover in the Celerra Network Server.

retention period: Period of time a file remains in the FLR state in the file system enabled for file-level retention. The Celerra administrator or a third-party archiving application manages a file's retention period by changing the file's last access time, by using NFS or CIFS, to a future time.

slice volume: On a Celerra system, 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: Automated Volume Management (AVM), a Celerra feature, organizes available disk volumes into groupings called storage pools. Storage pools are used to allocate available storage to Celerra file systems. Storage pools can be created automatically by AVM or manually by the user.

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

stripe volume: Arrangement of volumes that appear as a single volume. Allows for stripe units, which cut across the volume and are addressed in an interlaced manner. Stripe volumes make load balancing possible. *See also* disk volume, metavolume, stripe volume, and volume.

volume: On a Celerra system, 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.

Restrictions

When creating volumes on a Celerra Network Server attached to EMC Symmetrix® storage system, use standard Symmetrix volumes (also called hypervolumes), not Symmetrix metavolumes.

Cautions

This section lists the cautions for using this feature on the Celerra Network Server. If any of this information is unclear, contact your EMC Customer Support Representative for assistance.

- ◆ 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.
- ◆ Too many files in the root (/) of any file system might impact system performance. For optimal performance, the number of objects (such as files, subdirectories) should not exceed 500 names.
- ◆ Review ["Integration considerations" on page 19](#) if you intend to use file systems with Celerra features such as international character sets, EMC SnapSure™, EMC TimeFinder®/FS, quotas, or the Celerra AntiVirus Agent (CAVA).
- ◆ If you plan to set quotas on a file system to control the amount of space 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. *Using Quotas on EMC Celerra* provides additional information on quotas.
- ◆ If your user environment requires international character support (that is, support of non-English character sets or Unicode characters), configure your Celerra Network Server to support this feature before you create file systems. *Using International Character Sets with EMC Celerra* provides more information on international character support and how to configure it on the Celerra Network Server.
- ◆ If you plan to create TimeFinder/FS NearCopy and FarCopy snapshots of a production file system (PFS), do not use slice volumes (nas_slice) when you create the PFS. Instead, use the full disk presented to the Celerra Network Server. TimeFinder copies and restores volumes in their entirety and does not recognize sliced partitions created by the host (such as the Celerra Network Server).
- ◆ Do not attempt to use Symmetrix TimeFinder tools and utilities with file system copies created by Celerra 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 Celerra.

- ◆ 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 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 fsck utility on an unmounted file system, restart fsck utility on the unmounted file system.

Volumes and file system concepts

The Celerra Network Server 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 Celerra volumes and file systems without having to create and manage underlying volumes. AVM is a Celerra Network 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 EMC Celerra Volumes and File Systems with Automatic Volume Management* provides additional information on AVM capabilities on the Celerra Network Server.

File systems

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

- ◆ Expandable storage capacity that might be needed to dynamically expand a file system
- ◆ 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 Celerra Network Server creates different file systems based on how they are used. The file system types are listed in [Table 1 on page 7](#).

Table 1 Celerra file system types (page 1 of 2)

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 1 Celerra file system types (page 2 of 2)

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	NMFS

Inode

An inode is a data structure that stores information on files and the location of file blocks in the NFS file system. Celerra 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. The nbpi value reflects the expected average size of files in the file system. The nbpi value can be from 2048 bytes to 8192 bytes (default). EMC recommends an nbpi value of one inode for every 8192 bytes. The nbpi value can be changed only when you create a file system.

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.

There are several types of volumes. Each volume type provides different benefits to satisfy storage requirements. With the exception of system volumes, all volumes are initially available for data storage. The Celerra Network Server supports the volume types listed in [Table 2 on page 9](#).

Table 2 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 Celerra Network Server attached to a Symmetrix system.

["Volume configuration guidelines" on page 18](#) provides additional information on common volume configurations and considerations associated with each volume type.

Disk volumes

A Celerra disk volume is a physical storage unit as exported from the storage system to the Celerra Network Server. Disk volumes are the underlying storage of all other Celerra volume types.

A disk volume equates to a LUN as presented to the Celerra Network Server by the storage system. Each LUN is a usable storage-system volume that appears as a Celerra disk volume to the Celerra Network Server. Disk volumes are typically created by EMC support personnel during the initial installation and setup of the Celerra Network Server. 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. Specifying an offset is rarely used.

You must first identify the volume from which the slice volume will be created. The root slice volumes 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 CLARiiON® storage systems because of the larger LUNs presented to the Celerra Network Server.

Figure 1 on page 10 shows an 18 GB volume on which a 2 GB slice is defined.

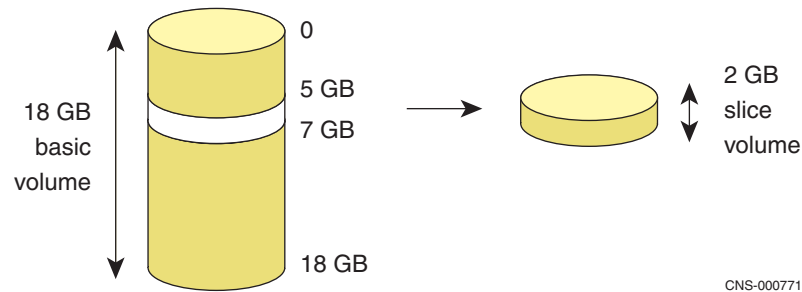


Figure 1 Slice volumes

A disk volume, stripe volume, or metavolume used as part of a business continuance volume (BCV) should not be sliced. "BCV" on page 13 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 11 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 10 shows an example of a stripe volume. The stripe is created across four participating volumes of equal size.

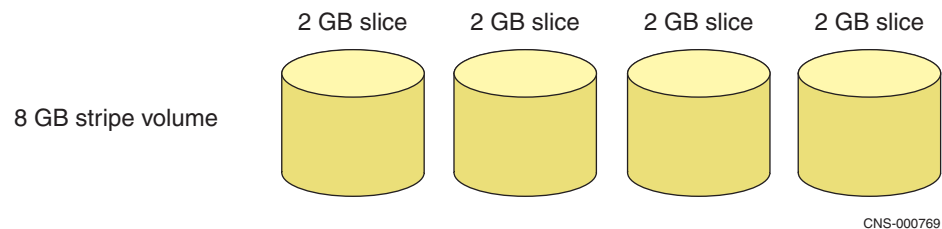


Figure 2 Stripe volumes

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 11](#) shows addressing within a stripe volume.

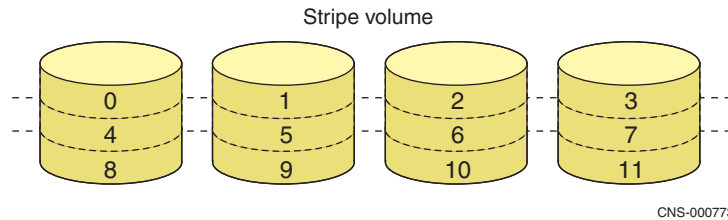


Figure 3 Addressing a stripe volume

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 19](#) 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.

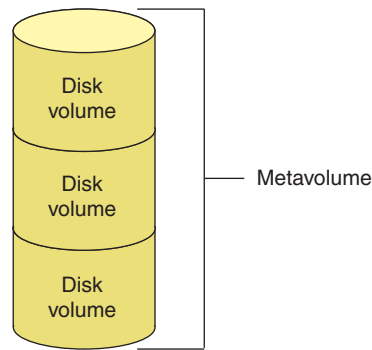
EMC recommends using only one type of volumes within a metavolume. For example, you create and combine stripe volumes, define them as a metavolume, and then create a file system with space from that metavolume.

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 a metavolume, the metavolume is automatically extended. [Figure 4 on page 12](#) shows a metavolume configuration that uses three disk volumes.

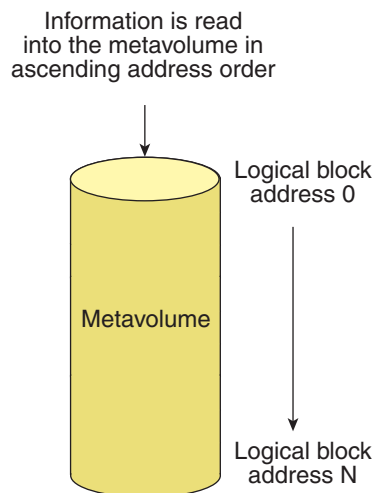


CNS-000774

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 12](#) shows metavolume addressing.



CNS-000770

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 13](#).

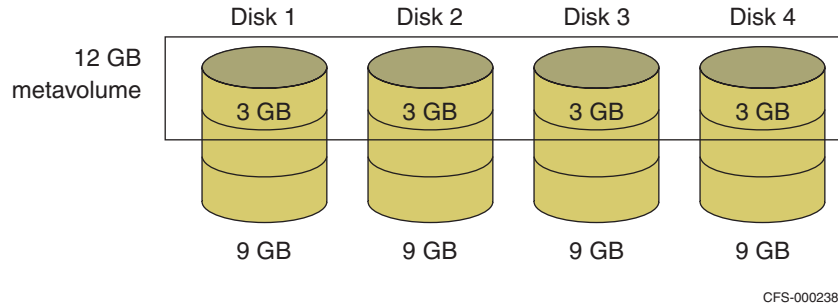


Figure 6 Metavolume created from slice volumes

Create striped volumes over a specified number of disk volumes defining a 32 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 13](#).

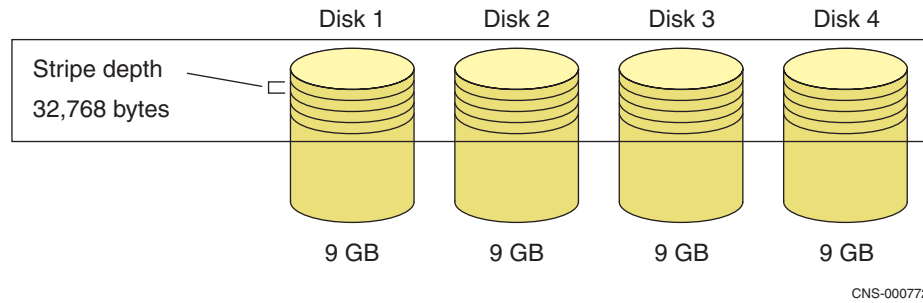


Figure 7 Metavolume created from stripe volumes

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 Celerra Network Server uses BCVs to create file system copies and mirror file systems dynamically. The EMC Customer Engineer creates BCVs on the storage system before installing the Celerra Network Server 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 14](#) shows the relationship between standard volumes and BCVs.

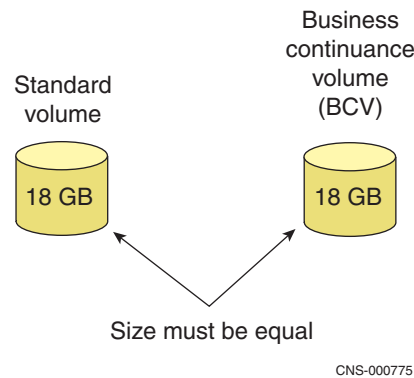


Figure 8 BCV

Note: In [Figure 8 on page 14](#), size refers to a disk volume's geometry.

BCVs are based on the LUN (entire disk volume) presented to the Celerra Network Server. 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.



CAUTION

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

System requirements for managing volumes and file systems manually

Table 3 on page 15 describes the Celerra Network Server software, hardware, network, and storage configurations required for using volumes and file systems as described in this document.

Table 3 Managing volumes and file systems manually system requirements

Software	Celerra Network Server version 5.6
Hardware	No specific hardware requirements
Network	No specific network requirements
Storage	Any Celerra-qualified storage system

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

Planning considerations for managing volumes and file systems manually

This section provides information that is helpful to plan file systems for the Celerra Network Server:

- ◆ ["Supported file system access protocols" on page 16](#)
- ◆ ["File system size guidelines" on page 17](#)
- ◆ ["NMFS" on page 17](#)
- ◆ ["Volume configuration guidelines" on page 18](#)
- ◆ ["Stripe volume configuration considerations" on page 19](#)
- ◆ ["Integration considerations" on page 19](#)

Supported file system access protocols

The Celerra Network Server supports access to file systems by using NFS, CIFS, FTP, TFTP, and MPFS 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 their normal manner.

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

- ◆ *Configuring NFS on EMC Celerra*
- ◆ *Configuring CIFS on EMC Celerra*
- ◆ *Managing EMC Celerra for the Windows Environment*
- ◆ *Managing EMC Celerra for a Multiprotocol Environment*

The Celerra Network Server 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 EMC Celerra User Mapping* provides additional information.

- ◆ *Using FTP on EMC Celerra Network Server*

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.

- ◆ *Using TFTP on EMC Celerra Network Server*

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.

- ◆ *Using MPFS on EMC Celerra*

A multi-path file system (MPFS) adds a thin, lightweight protocol known as the File Mapping Protocol (FMP) that works with Celerra protocols such as NFS and CIFS to control metadata operations. FMP exchanges file layout information and manages sharing conflicts between clients and Celerra Network Servers. The clients use the file layout information to read and write file data directly from and to the storage system.

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 Celerra Replicator™ feature must be less than the total supported capacity of the Data Mover.

For more information on file system size guidelines, refer to the E-Lab Interoperability Navigator available on [Powerlink](#).

NMFS

An NMFS allows you to manage a collection of file systems — 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 Celerra features that support NMFS are:

- ◆ SnapSure
- ◆ Replicator
- ◆ Quotas

- ◆ Security policies
- ◆ Backup
- ◆ EMC Symmetrix Remote Data Facility (SRDF®)
- ◆ TimeFinder/FS
- ◆ MPFS access to MPFS-enabled file systems
- ◆ CDMS
- ◆ FileMover
- ◆ EMC MirrorView™/Synchronous

Volume configuration guidelines

Volume configurations guidelines are helpful to plan the storage structure of file systems. [Table 4 on page 18](#) lists common volume configurations and considerations associated with each volume type.

Table 4 Volume configuration guidelines

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

Stripe volume configuration considerations

The guidelines to configure a stripe volume:

- ◆ The use of different stripe sizes depends on the applications you are using. The stripe depth must be entered in multiples of 8192 bytes. EMC recommends a stripe size of 32,768 bytes (default) for file systems that run in a CIFS or NFS environment with a Symmetrix or CLARiiON storage system. A 256 KB stripe size is recommended for MPFSi and MPFS environments, while RAID 3 and 64 KB stripe size are recommended for ATA-based file systems. For FLARE 24 and later, RAID 5 is recommended for ATA-based file systems with 64 KB stripe size.
- ◆ 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 20](#)
- ◆ ["TimeFinder/FS" on page 20](#)
- ◆ ["File-level retention" on page 20](#)
- ◆ ["SRDF" on page 21](#)
- ◆ ["MPFS" on page 21](#)
- ◆ ["Replicator/SnapSure" on page 21](#)
- ◆ ["Mirrorview/S" on page 21](#)

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 EMC Celerra 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 with EMC Celerra 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 until it is deleted. After you create and enable an FLR file system, you can apply FLR protection on each file. Files in the FLR state can be stored with retention periods, which prohibits the deletion of the file until expiration.

Using File-Level Retention on EMC Celerra provides additional information on FLR storage technology and file system behavior.

SRDF

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

MPFS

File systems mounted read-only are not acknowledged by clients that use MPFS, and thereby allow clients to write to the file system.

You cannot enable MPFS access to file systems with a stripe depth of less than 32 KB. *Using MPFS on EMC Celerra* provides additional information on MPFS.

Replicator/SnapSure

By using Celerra 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 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 EMC Celerra Replicator (V1)* and *Using EMC Celerra Replicator (V2)* contain instructions to recover from this situation. *Managing EMC Celerra Volumes and File Systems with Automatic Volume Management* provides information on automatic file system extension.

There must be sufficient file system space and disk storage available to support Replicator and SnapSure operations. To review the entire file system size, use `nas_fs -list`. To calculate the SavVol file size, use `nas_disk -size`. The *EMC Celerra Network Server Command Reference Manual* provides a detailed synopsis of the commands associated with SnapSure and Celerra Replicator.

Mirrorview/S

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

If you use the AVM feature to create the file systems, you must use the appropriate MirrorView AVM pools for your RAID configuration (`cm_r1`, `cm_r6`, `cm_r5_performance`, `cm_r5_recovery`, `cmata_archive`, `cmata_r3`, or `cmata_r6`).

User interface choices for managing volumes and file systems manually

The Celerra Network Server 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 Celerra management applications:

- ◆ Celerra Manager — Basic Edition
- ◆ Celerra Manager — Advanced Edition
- ◆ Celerra Monitor
- ◆ Microsoft Management Console (MMC) snap-ins
- ◆ Active Directory Users and Computers (ADUC) extensions

For additional information about managing your Celerra, refer to:

- ◆ *Learning about EMC Celerra on the EMC Celerra Network Server Documentation CD*
- ◆ *Celerra Manager online help*
- ◆ *Application's online help system on the EMC Celerra Network Server Documentation CD*

The *Installing EMC Celerra Management Applications* includes instructions on launching Celerra Manager, and on installing the MMC snap-ins and the ADUC extensions.

Roadmap for managing volumes and file systems manually

The tasks to configure Celerra volumes and file systems manually are:

1. ["Creating volumes" on page 24](#)
2. ["Creating file systems" on page 30](#)

The tasks to manage Celerra volumes and file systems manually are:

- ◆ ["Managing file systems" on page 36](#)
- ◆ ["Managing volumes" on page 58](#)
- ◆ ["Monitoring and repairing file systems" on page 64](#)

Creating volumes

When reserving CLARiiON 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.

The tasks to create and configure volumes manually are:

1. ["Determine storage availability" on page 24](#)
2. ["Creating volumes" on page 25](#)
3. ["Adding disk volumes to a Celerra gateway system with a CLARiiON array" on page 27](#)

Determine storage availability

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 Celerra Network Server 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_1disk	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							

List slice volumes

Before naming a slice volume, verify if the name is available.

Action

To view a list of slice volume names in use, type:

```
$ nas_slice -list
```

Output

This is a partial listing of the displayed slice volume list.

id	inuse	slice_of	offsetMB	sizeMB	name
1	n	1	0	134	root_dos
2	n	1	134	1	root_layout
3	y	1	135	16	root_slice_1
4	y	1	151	16	root_slice_2
5	y	1	167	16	root_slice_3
6	y	1	183	16	root_slice_4
7	y	1	199	16	root_slice_5
8	y	1	215	16	root_slice_6
9	y	1	231	16	root_slice_7
10	y	1	247	16	root_slice_8
11	y	1	263	16	root_slice_9
12	y	1	279	16	root_slice_10
13	y	1	295	16	root_slice_11
14	y	1	311	16	root_slice_12
15	y	1	327	16	root_slice_13
16	y	1	343	16	root_slice_14
17	y	1	359	16	root_slice_15
18	y	1	375	16	root_slice_16
19	n	1	391	4	root_rdf_channel
20	n	1	395	4	root_log_2
.					
.					
.					
121	y	115	1051855	5	s121
122	y	115	1051860	64	s122
124	y	115	1052124	200	s124

Creating volumes

There are several types of volumes. Each volume type provides different benefits to satisfy storage requirements. "[Volumes](#)" on page 9 provides detailed information on volume types. "[Volume configuration guidelines](#)" on page 18 lists common volume configurations and considerations associated with each volume type.

You can create three types of volumes:

- ◆ Slice volume
- ◆ Stripe volume
- ◆ Metavolume

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 19](#) 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> = size of the stripe volume in megabytes

<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
<p>To create a metavolume from a stripe volume, use this command syntax:</p> <pre>\$ nas_volume -name <name> -create -Meta [<volume_name>,...]</pre> <p>where:</p> <p><name> = name of the stripe volume</p> <p><volume_name> = names of the metavolumes separated by commas</p> <p>Example:</p> <p>To create metavolumes named slv1, slv2, and slv3 on disk volume d7, type:</p> <pre>\$ nas_volume -name mtv1 -create -Meta slv1, slv2, slv3</pre>
Output
<pre>id = 268 name = mtv1 acl = 0 in_use = False type = meta volume_set = slv1, slv2, slv3 disks = d8, d19, d9</pre>

Adding disk volumes to a Celerra gateway system with a CLARiiON array

If a Celerra gateway system is attached to a CLARiiON array and you want to add disk volumes to the configuration, use this procedure.

Before adding disk volumes to a Celerra gateway system with a CLARiiON array

EMC Celerra system network servers that are gateway network-attached storage systems and that connect to EMC Symmetrix and CLARiiON arrays are:

- ◆ Celerra NS500G
- ◆ Celerra NS500GS
- ◆ Celerra NS600G
- ◆ Celerra NS600GS
- ◆ Celerra NS700G
- ◆ Celerra NS700GS
- ◆ Celerra NS704G
- ◆ Celerra NS40G
- ◆ Celerra NS-G8

A Celerra gateway system stores data on CLARiiON user LUNs or Symmetrix hypervolumes. If the user LUNs or hypervolumes are not configured correctly on the array, Celerra AVM and Celerra Manager cannot be used to manage the storage. The initial setup of disk volumes on these gateway storage systems is typically performed by EMC support personnel.

In this task:

1. Use EMC Navisphere® Manager or the EMC Navisphere CLI to create CLARiiON user LUNs.
2. Use Celerra Manager to make the new user LUNs available to the Celerra Network Server as disk volumes.

The user LUNs must be created before you create file systems.

Note: To add CLARiiON user LUNs, you must be familiar with EMC Navisphere Manager or the EMC Navisphere CLI and the process of creating RAID groups and CLARiiON user LUNs for the Celerra volumes. The documentation for EMC Navisphere Manager and EMC Navisphere CLI, available on EMC [Powerlink](#), describes how to create RAID groups and user LUNs.

Add CLARiiON user LUNs and make them available to Celerra gateway system

Step	Action
1.	<p>By using EMC Navisphere Manager or EMC Navisphere CLI, create RAID groups and user LUNs as needed for Celerra volumes. Ensure that:</p> <ul style="list-style-type: none"> You add the LUNs to the gateway system's storage group. You 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. For Fibre Channel disks, the paired LUNs do not have to be in the same RAID group. For RAID 5 on Fibre Channel disks, the RAID group must use five or nine disks. RAID 1 groups always use two disks. RAID 6 groups have no restrictions on the number of disks. For ATA disks, all LUNs in a RAID group must belong to the same SP. Create pairs by using LUNs from two RAID groups. ATA disks must be configured as RAID 5, RAID 6, or RAID 3. The host LUN identifier (HLU) is greater than or equal to 16 for user LUNs. Settings to use when creating the user LUNs: <ul style="list-style-type: none"> RAID Type: RAID 5, RAID 6, or RAID 1 for FC disks and RAID 5, RAID 6, or RAID 3 for ATA disks LUN ID: Select the first available value Element Size: 128 ... 256 MB 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 2 TB <p>Note: If you are creating 4+1 RAID 3 LUNs, the Number of LUNs to Bind value is 1.</p> <ul style="list-style-type: none"> When you add the LUN to the storage group for a gateway system, set the HLU to 16 or greater.
2.	<p>Make the new user LUNs available to the Celerra system by using Celerra Manager.</p> <ol style="list-style-type: none"> Open the Storage System page for the Celerra system (Storage > Systems). Click Rescan. <p>Note: Do not change the HLU of the Celerra LUNs after rescanning.</p>

Creating file systems

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

The tasks to create a file system are:

1. ["Create a file system" on page 31](#)
2. ["\(Optional\) Creating a mount point" on page 31](#)
3. ["Mount a file system" on page 32](#)
4. ["\(Optional\) Create an NMFS" on page 34](#)

Before creating a file system

A file system name must be unique on a particular Celerra Network Server. It can contain up to 240 ASCII characters, including upper and lowercase letters, numbers, hyphens (-), underscores (_), and periods (.). Alpha-numeric names are accepted.

A file system name cannot:

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

Create a file system

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 with existing volumes called ufs1, type:

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

Output

```
id          = 18
name        = ufs1
acl         = 0
in_use      = False
type        = ufs
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) Creating 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.

Before creating a mount point

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). International characters are not supported as mount point names.

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.

Create a mount point on a Data Mover

Action
<p>To create a mount point on a Data Mover, use this command syntax:</p> <pre>\$ server_mountpoint <movername> -create <pathname></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><pathname> = path to mount point</p> <p>Example:</p> <p>To create a mount point named server_3, type:</p> <pre>\$ server_mountpoint server_3 -create /ufs1</pre>
Output
<pre>server_3: done</pre>

Mount a file system

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

Before mounting a file system

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 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.

Mount a file system on a Data Mover

The `-option` argument is used to specify a number of mount options. The *EMC Celerra Network Server Command Reference Manual* provides a complete list of mount options available.

Action
<p>To mount a file system on a mount point that is on a Data Mover, use this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> <mount_point></pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover<options> = list of mount options separated by comma<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 (/) <p>Example:</p> <p>To mount ufs1 on mount point /ufs1 with access checking policy set to NATIVE and nooplock turned on, type:</p> <pre>\$ server_mount server_2 -option accesspolicy=NATIVE, nooplock ufs1 /ufs1</pre>
Output
<pre>server_2: done</pre>

(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 54](#) explains how to unmount a file system from a Data Mover.

Step	Action
1.	<p>Create an NMFS by using this command syntax:</p> <pre>\$ nas_fs -name <name> -type nmfs -create</pre> <p>where:</p> <p><name> = name of the NMFS</p> <p>Example:</p> <p>To create an NMFS named <code>nmfs1</code>, type:</p> <pre>\$ nas_fs -name nmfs1 -type nmfs -create</pre> <p>Output:</p> <pre>id = 26 name = nmfs1 acl = 0 in_use = False type = nmfs worm = off volume = 0 pool = rw_servers= ro_servers= rw_vdms = ro_vdms = stor_devs = disks =</pre> <p>Note: There is no volume allocation.</p>
2.	<p>Create a mount point in the root of the designated Data Mover for the new file system.</p>
3.	<p>Mount the NMFS as read-only on the Data Mover by using this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> <mount_point></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = list of mount options separated by comma</p> <p><fs_name> = name of the NMFS</p> <p><mount_point> = path to mount point for the Data Mover; a <mount_point> must begin with a forward slash (/)</p> <p>Example:</p> <p>To mount an NMFS named <code>nmfs1</code> as read-only on <code>server_3</code>, type:</p> <pre>\$ server_mount server_3 -option ro nmfs1 /nmfs1</pre> <p>Output:</p> <pre>server_3: done</pre>

Step	Action
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.

Step	Action
1.	Create a volume for the component file system.
2.	<p>Create the component file system on the new volume by using this command syntax:</p> <pre>\$ nas_fs -name <name> -create <volume_name></pre> <p>where:</p> <p><name> = name assigned to the file system</p> <p><volume_name> = name of the volume</p> <p>Example:</p> <p>To create a component file system called ufs1 on volume mtv1, type:</p> <pre>\$ nas_fs -name ufs1 -create mtv1</pre>
3.	<p>Mount the component file system to the NMFS by using this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> <mount_point></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = list of mount options separated by comma</p> <p><fs_name> = name of the NMFS</p> <p><mount_point> = pathname of the NMFS which is in the format /nmfs path/component file system name</p> <p>Example:</p> <p>To mount ufs1, as a part of the NMFS nmfs1, type:</p> <pre>\$ server_mount server_2 ufs1 /nmfs1/ufs1</pre> <p>Output:</p> <pre>server_2: done</pre>

Managing file systems

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

The tasks to manage file systems are:

- ◆ ["Export a file system" on page 36](#)
- ◆ ["List file systems" on page 38](#)
- ◆ ["View the file system configuration information" on page 39](#)
- ◆ ["List mount points" on page 39](#)
- ◆ ["List mounted file systems" on page 40](#)
- ◆ ["Check disk space capacity for a single mounted file system" on page 40](#)
- ◆ ["Check inode capacity" on page 41](#)
- ◆ ["Extend a file system" on page 43](#)
- ◆ ["Extend replicated file systems" on page 46](#)
- ◆ ["Adjust file system size threshold" on page 48](#)
- ◆ ["Add an existing file system to the NMFS" on page 49](#)
- ◆ ["Move an NMFS" on page 51](#)
- ◆ ["Rename a file system" on page 51](#)
- ◆ ["Enhance file read/write performance" on page 52](#)
- ◆ ["Unmount all file systems from a Data Mover" on page 54](#)
- ◆ ["Delete a file system" on page 55](#)

Export a file system

The tasks to export a file system are:

- ◆ ["Export a file system from a Data Mover for NFS access" on page 37](#)
- ◆ ["Export a file system from a Data Mover for CIFS access" on page 37](#)

Before exporting 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 only be accessed by using NFSv4

For CIFS clients, the component file system is shared.

Export a file system from a Data Mover for NFS access

Action
<p>To export a file system for NFS access, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -option <options>/<pathname></pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover<options> = NFS export options<pathname> = pathname of the file system to mount <p>Example:</p> <p>To export the file system ufs2 on an NFS client, type:</p> <pre>\$ server_export server_3 -Protocol nfs -option root=10.1.1.1 /ufs2</pre>
Output
<pre>server_3: done</pre>

Export a file system from a Data Mover for CIFS access

Action
<p>To export a file system for CIFS access, use this command syntax:</p> <pre>\$ server_export <movername> -Protocol cifs -name <sharename>/<pathname></pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover<sharename> = name of the shared component file system<pathname> = pathname of the file system to mount <p>Example:</p> <p>To export the file system ufs2 on a CIFS client, type:</p> <pre>\$ server_export server_3 -Protocol cifs -name ufs2 /ufs2</pre>
Output
<pre>server_3: done</pre>

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 only export 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 on a Celerra Network Server, 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 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         = ufs
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
<p>To display a list of all file systems mounted on a Data Mover, use this command syntax:</p> <pre>\$ server_mount <movername></pre> <p>where: <movername> = name of the Data Mover</p> <p>Example:</p> <p>To list all file systems mounted on server_3, type:</p> <pre>\$ server_mount server_3</pre>
Output
<pre>server_3: fs2 on /fs2 udfs,perm,rw fs1 on /fs1 udfs,perm,rw root_fs_3 on / udfs,perm,rw</pre>

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
<p>To display disk space of a file system, use this command syntax:</p> <pre>\$ nas_fs -size <fs_name></pre> <p>where: <fs_name> = name of the file system</p> <p>Example:</p> <p>To view the total space available on ufs1, type:</p> <pre>\$ nas_fs -size ufs1</pre>
Output
<pre>total = 2041959 avail = 2041954 used = 6 (0%)(sizes in MB) (blockcount = 4246732800) volume: total = 2073600 (sizes in MB) (blockcount = 4246732800) root_fs_3 on / udfs,perm,rw</pre>

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

This section describes how to check the inode capacity of a file system on a Data Mover.

Before checking the inode capacity

The number of inodes defines the number of filenames and directory names 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.

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

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.

Before extending a file system

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 39](#) provides the information to determine whether a file system has an associated pool. *Managing EMC Celerra Volumes and File Systems with Automatic Volume Management* provides additional information on extending file systems created with AVM.

Extend a file system

Step	Action
1.	<p>Check the size of the file system before extending it by using this command syntax:</p> <pre>\$ nas_fs -size <fs_name></pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p>Example:</p> <p>To check the size of the file system ufs1, type:</p> <pre>\$ nas_fs -size ufs1</pre> <p>Output:</p> <pre>total = 67998 avail = 67997 used = 0 (0%) (sizes in MB) volume: total = 69048 (sizes in MB)</pre>
2.	<p>Extend the file system by using this command syntax:</p> <pre>\$ nas_fs -xtend <fs_name> <volume_name></pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p><volume_name> = name of the volume</p> <p>Example:</p> <p>To extend the file system ufs1, type:</p> <pre>\$ nas_fs -xtend ufs1 emtv2b</pre> <p>Output:</p> <pre>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=c0t3l8-15-0 server=server_2 disk=d4 symm_dev=002804000190-0035 addr=c0t3l9-15-0 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</pre>

Step	Action
3.	<p>Check the size of the file system after extending it by using this command syntax:</p> <pre>\$ nas_fs -size <fs_name></pre> <p>where: <fs_name> = name of the file system</p> <p>Example:</p> <p>To check the size of the file system ufs1 after extending it, type:</p> <pre>\$ nas_fs -size ufs1</pre> <p>Output:</p> <pre>total = 138096 avail = 138096 used = 0 (0%) (sizes in MB) volume: total = 138096 (sizes in MB)</pre>

Extend replicated file systems

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

Before extending replicated file systems

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 *EMC Celerra Network Server Command Reference Manual* provides a detailed synopsis of the commands associated with the Celerra Replicator.
- ◆ Verify whether there is enough volume space to extend the source and destination file systems.

Extend source file system and corresponding destination file system

Step	Action												
1.	<p>On the primary site, verify the current sizes of the source and destination file systems by using this command syntax:</p> <pre>\$ nas_fs -size <fs_name></pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p>Example:</p> <p>To verify the current size of the source file system src_ufs1, type:</p> <pre>\$ nas_fs -size src_ufs1</pre> <p>Output:</p> <pre>total = 67998 avail = 67997 used = 0 (0%) (sizes in MB) volume: total = 69048 (sizes in MB)</pre>												
2.	<p>On the remote site, verify the current sizes of the source and destination file systems by using this command syntax:</p> <pre>\$ server_df <fs_name></pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p>Example:</p> <p>To verify the current size of the source and destination file system dst_ufs1, type:</p> <pre>\$ server_df server_2 dst_ufs1</pre> <p>Output:</p> <table><tr><th>Filesystem</th><th>kbytes</th><th>used</th><th>avail</th><th>capacity</th><th>Mounted on</th></tr><tr><td>ppfs1</td><td>4130288</td><td>97392</td><td>4032896</td><td>2%</td><td>/pfs1</td></tr></table>	Filesystem	kbytes	used	avail	capacity	Mounted on	ppfs1	4130288	97392	4032896	2%	/pfs1
Filesystem	kbytes	used	avail	capacity	Mounted on								
ppfs1	4130288	97392	4032896	2%	/pfs1								

Step	Action
3.	<p>Extend the source file system (on the primary site) by using this command syntax:</p> <pre>\$ nas_fs -xtend <fs_name> <volume_name></pre> <p>where:</p> <p><fs_name> = name of the source file system</p> <p><volume_name> = name of the volume</p> <p>Example:</p> <p>To extend the source file system (on the primary site), type:</p> <pre>\$ nas_fs -xtend src_ufs1 mtv4</pre>
4.	<p>Check the size of the file system after extending it by using this command syntax:</p> <pre>\$ nas_fs -size <fs_name></pre> <p>where:</p> <p><fs_name> = name of the source file system</p> <p>Example:</p> <p>To check the size of the source file system src_ufs1 after extending it, type:</p> <pre>\$ nas_fs -size src_ufs1</pre> <p>Output:</p> <pre>total = 138096 avail = 138096 used = 0 (0%) (sizes in MB) volume: total = 138096 (sizes in MB)</pre>

Adjust file system size threshold

The tasks to adjust file system size threshold are:

- ♦ ["Adjust the file system size threshold for all file systems on the Celerra Network Server" on page 48](#)
- ♦ ["Adjust the file system size threshold on a single Data Mover" on page 49](#)

Before adjusting file system size threshold

Because file system performance can degrade as its used space approaches 100 percent full, the Celerra Network Server 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 EMC Celerra Events and Notifications*.

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.

Adjust the file system size threshold for all file systems on the Celerra Network Server

Step	Action
1.	<p>Change the size threshold for all file systems by using this command syntax:</p> <pre>\$ server_param ALL -facility <facility_name> -modify <param_name> -value <new_value></pre> <p>where:</p> <ul style="list-style-type: none"><movernam> = 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 <p>Example:</p> <p>To change the size threshold for all file systems to 85%, type:</p> <pre>\$ server_param ALL -facility file -modify fsSizeThreshold -value 85</pre>
2.	<p>Restart the Data Movers by using this command syntax:</p> <pre>\$ server_cpu ALL -reboot now</pre>

Adjust the file system size threshold on a single Data Mover

Step	Action
1.	<p>Change the file system size threshold on a single Data Mover by using this command syntax:</p> <pre>\$ server_param <movername> -facility <facility_name> -modify <param_name> -value <new_value></pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover<facility_name> = facility for the parameters<param_name> = name of the parameter<new_value> = new value for the parameter <p>Example:</p> <p>To change the size threshold for all file systems on server_2, type:</p> <pre>\$ server_param server_2 -facility file -modify fsSizeThreshold -value 85</pre>
2.	<p>Restart the Data Mover for the change to take effect by using this command syntax:</p> <pre>\$ server_cpu <movername> -reboot now</pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover <p>Example:</p> <p>To restart server_2 for the change to take effect, type:</p> <pre>\$ server_cpu server_2 -reboot now</pre>

Add an existing file system to the 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.

Step	Action
1.	<p>Permanently unmount the file system from a Data Mover by using this command syntax:</p> <pre>\$ server_umount <movername> -perm <fs_name></pre> <p>where:</p> <ul style="list-style-type: none"><movername> = name of the Data Mover<fs_name> = name of the file system to unmount <p>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.</p> <p>Example:</p> <p>To permanently unmount a file system named fs1, type:</p> <pre>\$ server_umount server_2 -perm /fs1</pre> <p>Output:</p> <pre>server_2: done</pre>
2.	Create a new mount point for the file system in the NMFS.

Step	Action
3.	<p>Mount the file system in the NMFS by using this command syntax:</p> <pre>\$ server_mount <movername> -option <options> <fs_name> <mount_point></pre> <p>where:</p> <ul style="list-style-type: none"> <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 <p>Example:</p> <p>To mount a file system on a mount point on server_3 with a nolock option, type:</p> <pre>\$ server_mount server_3 -option nolock fs5/nmfs4/fs5</pre> <p>Output:</p> <pre>server_2: done</pre>

Move an NMFS

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

Step	Action
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
<p>To rename a file system, use this command syntax:</p> <pre>\$ nas_fs -rename <old_name> <new_name></pre> <p>where:</p> <p><old_name> = existing name of the file system</p> <p><new_name> = new name of the file system</p> <p>Example:</p> <p>To rename a file system ufs as ufs1, type:</p> <pre>\$ nas_fs -rename ufs ufs1</pre>
Output
<pre>id = 18 name = ufs1 acl = 0 in_use = False type = udfs 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</pre>

Enhance file read/write performance

The Celerra Network Server 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 52](#)
- ◆ ["Turn off read prefetch for all file systems on a Data Mover" on page 53](#)
- ◆ ["Enable uncached write mechanism" on page 54](#)

Turn off read prefetch for a specific file system

Action
<p>To turn off the read prefetch mechanism, use this command syntax:</p> <pre>\$ server_mount <movername> -option <options>,noprefetch <fs_name> <mount_point></pre> <p>where:</p> <ul style="list-style-type: none"><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 (/) <p>Example:</p> <p>To turn off the read prefetch mechanism for ufs1, type:</p> <pre>\$ server_mount server_3 -option rw,noprefetch ufs1 /ufs1</pre>
Output
<pre>server_3: done</pre>

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

Step	Action
1.	<p>Turn off the read prefetch mechanism for all file systems on a Data Mover by using this command syntax:</p> <pre>\$ server_param <movername> -facility <facility_name> -modify prefetch -value 0</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><facility_name> = facility for the parameters</p> <p>Example:</p> <p>To turn off the prefetch mechanism for all file systems on server_2, type:</p> <pre>\$ server_param server_2 -facility file -modify prefetch -value 0</pre>
2.	<p>Restart the Data Mover by using this command syntax:</p> <pre>\$ server_cpu <movername> -reboot</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To restart server_2 immediately, type:</p> <pre>\$ server_cpu server_2 -reboot now</pre>

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
<p>To turn on the uncached write mechanism for a file system, use this command syntax:</p> <pre>\$ server_mount <movername> -option <options>,uncached <fs_name> <mount_point></pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p><options> = specifies mount options, separated by commas</p> <p><fs_name> = name of the file system</p> <p><mount_point> = path to mount point for the Data Mover; a <mount_point> must begin with a forward slash (/)</p> <p>Example:</p> <p>To turn on the uncached write mechanism for the file system ufs1, type:</p> <pre>\$ server_mount server_3 -option rw,uncached ufs1 /ufs1</pre>
Output
<pre>server_3: done</pre>

Unmount all file systems from a Data Mover

The tasks to unmount all file systems are:

- ◆ ["Unmount all file systems temporarily" on page 55](#)
- ◆ ["Unmount all file systems permanently" on page 55](#)

Before unmounting 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.

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
<p>To temporarily unmount all file systems on a Data Mover, use this command syntax:</p> <pre>\$ server_umount <movername> -temp -all</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To temporarily unmount all file systems on server_2, type:</p> <pre>\$ server_umount server_2 -temp -all</pre>
Output
<pre>server_2: done</pre>

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
<p>To permanently unmount all file systems on a Data Mover, use this command syntax:</p> <pre>\$ server_umount <movername> -perm -all</pre> <p>where:</p> <p><movername> = name of the Data Mover</p> <p>Example:</p> <p>To permanently unmount all file systems on server_2, type:</p> <pre>\$ server_umount server_2 -perm -all</pre>
Output
<pre>server_2: done</pre>

Delete a file system

This section describes how to delete a file system or NMFS.

Before deleting a file system

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 with EMC Celerra* 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 54](#) provides additional information on unmounting file systems permanently.

Delete a file system or NMFS

Step	Action
1.	Back up all the data you want to keep.
2.	<p>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:</p> <pre>\$ nas_fs -info <fs_name></pre> <p>where: <fs_name> = name of the file system</p> <p>If the pool output line displays a value, the file system has an associated storage pool.</p> <p>Note: 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.</p>
3.	<p>Determine and notate the metavolume name on which the file system is built. You need to provide the metavolume name in step 10:</p> <pre>\$ nas_fs -info <fs_name></pre> <p>where: <fs_name> = name of the file system</p> <p>Note: The Volume field contains the metavolume name. The Disks field lists the disks providing storage to the file system.</p>
4.	If the file system has associated checkpoints, permanently unmount and then delete the checkpoints and their associated volumes.
5.	<p>If the file system has associated BCVs, break the connection between (unmirror) the file system and its BCVs.</p> <p><i>Using TimeFinder/FS, NearCopy, and FarCopy with EMC Celerra</i> provides information on how to unmirror a BCV.</p>
6.	<p>If the file system is an NFS-exported file system, permanently disable client access to the file system by using this command syntax:</p> <pre>\$ server_export <movername> -Protocol nfs -unexport -perm <pathname></pre> <p>where: <movername> = name of the Data Mover <pathname> = NFS entry</p>

Step	Action
7.	<p>If the file system is a CIFS-exported file system, permanently disable client access to the file system by using this command syntax:</p> <pre>\$ server_export <movername> -Protocol cifs -unexport <sharename></pre> <p>where: <movername> = name of the Data Mover <sharename> = name of the shared component file system</p>
8.	<p>Permanently unmount the file system from its associated Data Movers by using this command syntax:</p> <pre>\$ server_umount <movername> -perm <fs_name></pre> <p>where: <movername> = name of the Data Mover <fs_name> = name of the file system</p> <p>Note: To delete an NMFS, permanently unmount all component file systems in the NMFS.</p>
9.	<p>Delete the file system or NMFS from the Celerra Network Server by using this command syntax:</p> <pre>\$ nas_fs -delete <fs_name></pre> <p>where: <fs_name> = name of the file system</p> <p>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.</p> <p>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.</p>
10.	<p>Delete the metavolume on which the file system was created by using this command syntax:</p> <pre>\$ nas_volume -delete <volume_name></pre> <p>where: <volume_name> = name of the volume</p>
11.	<p>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:</p> <pre>\$ nas_volume -delete <volume_name></pre> <p>where: <volume_name> = name of the volume</p>
12.	<p>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:</p> <pre>\$ nas_volume -delete <volume_name></pre> <p>where: <volume_name> = name of the volume</p>
13.	<p>After freeing disk space, check for slice volumes, stripe volumes, and metavolumes not in use (n in the inuse column) by using these commands:</p> <pre>\$ nas_volume -list \$ nas_slice -list</pre> <p>Delete unused volumes until you free all the disk space you want.</p>

Managing volumes

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

The tasks to manage Celerra volumes are:

- ◆ ["List volumes" on page 58](#)
- ◆ ["Check volume capacity" on page 59](#)
- ◆ ["Rename a volume" on page 59](#)
- ◆ ["Clone a volume" on page 60](#)
- ◆ ["Delete a metavolume or stripe volume" on page 61](#)
- ◆ ["Delete a slice volume" on page 62](#)

List volumes

Action						
To list all volumes on a Celerra Network Server, type: \$ nas_volume -list						
Output						
This is a partial listing of the volume table 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

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

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
<p>To clone a volume, use this command syntax:</p> <pre>\$ nas_volume -Clone <volume_name> -option disktype=<disktype> <source_volume>:<destination_volume>,...</pre> <p>where:</p> <p><volume_name> = name of the volume to be cloned</p> <p><disktype> = type of disk to be created</p> <p><source_volume> = sets a specific disk volume set for the source volume</p> <p><destination_volume> = sets a specific disk volume set for the destination volume</p> <p>Example:</p> <p>To clone the metavolume mtv1, type:</p> <pre>\$ nas_volume -Clone mtv1</pre>
Output
<pre>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</pre>
Note
<p>The example clones the metavolume mtv1. The default name of the cloned metavolume is v128.</p>

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 61](#)
- ◆ ["Delete a metavolume" on page 61](#)

List metavolume information

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

Action
<p>To list the volume information, use this command syntax:</p> <pre>\$ nas_volume -info <volume_name></pre> <p>where:</p> <p><volume_name> = name of the metavolume</p> <p>Example:</p> <p>To list the volume information for <code>mtv1</code>, type:</p> <pre>\$ nas_volume -info mtv1</pre>
Output
<pre>id = 247 name = mtv1 acl = 0 in_use = False type = meta volume_set = stv1 disks = d3,d4,d5,d6</pre>
Note
<p>The <code>in_use</code> parameter for the <code>mtv1</code> metavolume is <code>False</code>, indicating that the metavolume is not in use by a file system.</p>

Delete a metavolume

Remove all file systems from a volume 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
<p>To delete a metavolume, use this command syntax:</p> <pre>\$ nas_volume -delete <volume_name></pre> <p>where:</p> <p><volume_name> = name of the metavolume to delete</p> <p>Example:</p> <p>To delete a metavolume named <code>mtv1</code>, type:</p> <pre>\$ nas_volume -delete mtv1</pre>

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 62](#)
- ◆ ["Delete a slice volume" on page 63](#)

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	
<p>To list the volume information, use this command syntax:</p> <pre>\$ nas_slice -info <slice_name></pre> <p>where:</p> <p><slice_name> = name of the slice volume</p> <p>Example:</p> <p>To list the slice volume information for slv1, type:</p> <pre>\$ nas_slice -info slv1</pre>	
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.

Monitoring and repairing file systems

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

- ◆ ["Run file system check" on page 65](#)
- ◆ ["Start an ACL check on the file system" on page 66](#)
- ◆ ["List file system checks" on page 67](#)
- ◆ ["Display the file system check information on a file system" on page 67](#)
- ◆ ["Display information on all current file system checks" on page 68](#)

Before monitoring and repairing a file system

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 5](#) 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, 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 *EMC Celerra Network Server Parameters Guide* provides detailed information on parameters.

Run file system check

This section describes how to start and monitor fsck on a file system.

Before running file system check

When the Celerra Network Server 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 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.

Start and monitor fsck on a file system

Action
<p>To start and monitor fsck on a file system, use this command syntax:</p> <pre>\$ nas_fsck -start <fs_name> -monitor</pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p>Example:</p> <p>To start fsck on ufs1 and monitor the progress, type:</p> <pre>\$ nas_fsck -start ufs1 -monitor</pre>
Output
<pre>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</pre>

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
<p>To start ACL check on a specified file system, use this command syntax:</p> <pre>\$ nas_fsck -start <fs_name> -aclchkonly</pre> <p>where:</p> <p><fs_name> = name of the file system</p> <p>Example:</p> <p>To start ACL check on ufs1 and monitor the progress, type:</p> <pre>\$ nas_fsck -start ufs1 -aclchkonly</pre>
Output
<pre>ACLCHK: in progress for file system ufs1</pre>

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 currently running, use this command syntax:

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

Example:

To display information on all file system checks 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
```

Troubleshooting volumes and file systems

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, please contact your EMC Customer Support Representative.

Where to get help

To obtain EMC support, product, and licensing information:

Product information – For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink website (registration required) at:

<http://Powerlink.EMC.com>

Technical support – For technical support, go to EMC Customer Service on Powerlink. After logging in to the EMC [Powerlink](#) website, go to **Support > Request Support**. To open a service request through Powerlink, you must have a valid support agreement. Please contact your EMC Customer Support Representative for details about obtaining a valid support agreement or to answer any questions about your account.

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

Problem Resolution Roadmap for EMC Celerra contains additional information about using [Powerlink](#) and resolving problems.

Known problems and limitations

[Table 5 on page 70](#) describes known problems that might occur when using volumes and file systems and presents workarounds.

Table 5 Volume and file system problems and workarounds

Known problem	Symptom	Workaround
Celerra Manager 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 on Celerra Manager, or directly on 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 <code>server_mount -all</code> 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.
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 you want to slice has enough unused space, use this command syntax: <code>\$ nas_volume -size <volume_name></code>

Error messages for volumes and file systems

As of version 5.6, all new event, alert, and status messages provide detailed information and recommended actions to help you troubleshoot the situation.

To view message details, use any of these methods:

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

Related information

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

- ◆ *Configuring CIFS on EMC Celerra*
- ◆ *Configuring NFS on EMC Celerra*
- ◆ *EMC Celerra Network Server Command Reference Manual*
- ◆ *EMC Celerra Network Server Parameters Guide*
- ◆ *Managing EMC Celerra for a Multiprotocol Environment*
- ◆ Online Celerra man pages

Celerra volumes and file systems information that is related, but is beyond the scope of this document, is included in *Managing EMC Celerra Volumes and File Systems with Automatic Volume Management*.

The *EMC Celerra Network Server Documentation CD*, supplied with your Celerra Network Server and also available on [Powerlink](#), provides general information on other Celerra publications.

Customer training programs

EMC customer training programs are designed to help you learn how EMC storage products work together and integrate within your environment to maximize your entire infrastructure investment. EMC customer training programs feature online and hands-on training in state-of-the-art labs conveniently located throughout the world. EMC customer training programs are developed and delivered by EMC experts. For program information and registration, log in to [Powerlink](#), our customer and partner website and select the Training menu.

Appendix A: GID support

The Celerra Network Server 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 Celerra command to verify whether a file system supports 16-bit or 32-bit GIDs.

Index

A

- adjusting file system size threshold 48, 49
- automatic file system extension 21
- Automatic Volume Management (AVM) 3
- AVM. See Automatic Volume Management (AVM) 3

B

- BCV. See business continuance volume (BCV) 3
- business continuance volume (BCV)
 - configuration 13
 - definition 3

C

- capacity
 - volume 59
- cautions
 - fsck 6
 - fsck processes 6
 - nas_db database 5
 - slice volumes 5
 - spanning storage systems 5
- CAVA
 - integration considerations 5
- Celerra Replicator 21
- Celerra volumes
 - BCV 13
 - cloning 60
 - disk 9
 - metavolume 12
 - slice 9
 - stripe 10
- cloning a volume 60
- component file system 17, 37
 - creating 35
 - definition 3
 - extending 43
 - unmounting 54
- creating
 - stripe volume 26, 27
 - volumes 24

D

- Data Mover
 - listing
 - mount points 39
 - mounted file systems 40
- deleting
 - file system 56
 - metavolume 61
 - slice volume 62
 - stripe volume 61
- disk volume
 - definition 3
 - explanation 9

- freeing file system space 56
- renaming 59
- unused 24

E

- exporting 37

F

- file system
 - concepts 7
 - definition 4
 - deleting 56
 - displaying mounted 40
 - freeing disk space allocated 56
 - mirroring 14
 - permanent
 - mount 32
 - quotas 5, 20
 - size guidelines 17
 - unmount all
 - permanent 55
 - temporary 55
- file system size threshold 48, 49
 - change
 - for Data Mover 48, 49
 - fsSizeThreshold parameter 48, 49
- file-level retention
 - definition 4
 - integration considerations 20
- FLR state
 - definition 4
- fsck
 - caution 6
 - repairing file systems 64
 - running file system check 65
 - starting an ACL check 65
 - system performance 6
- fsSizeThreshold parameter 48, 49

G

- GID support
 - restrictions 73

I

- inode
 - definition 4
- integration considerations
 - file-level retention 20
 - quotas 20
 - Replicator 21
 - SnapSure 21
 - SRDF 21
 - TimeFinder/FS 20
- International character support 5

L

- logical unit number (LUN)

definition 4
LUN. See logical unit number (LUN) 4

M

metavolume 11
 addressing 12
 concepts 7
 configuration guidelines 12
 creating 27
 definition 4
 deleting 61
 renaming 59
mirrored file system 14
mount point
 creating 31
 listing 39
MPFS. See multi-path file system (MPFS) 17
multi-path file system (MPFS) 17
 integration considerations 21

N

nas_db database
 cautions 5
nas_fsck 66, 67, 68
nested mount file system (NMFS)
 about 17
 definition 4
nested mount file system root
 definition 4
NMFS. See nested mount file system (NMFS) 4

P

parameters
 fsSizeThreshold 48, 49
protocol
 multi-path file system (MPFS) 17
 Trivial File Transfer Protocol (TFTP) 17

Q

quotas
 cautions 5
 integration considerations 5, 20

R

renaming
 disk volume 59
 metavolume 59
 slice volume 59
 stripe volume 59
Replicator
 integration considerations 21
restrictions
 Celerra file systems 5
 GID support 73
 nas_db database 5
 TimeFinder/FS 5

S

server_mount command 31, 52, 54
slice volume
 definition 4
 deleting 62
 how it works 9
 renaming 59
SnapSure
 integration considerations 5, 21
SRDF
 integration considerations 21
storage pool
 definition 4
storage system
 definition 4
stripe volume
 creating 26, 27
 definition 4
 deleting 61
 explanation 10
 improving performance with 11
 renaming 59

T

TFTP. See Trivial File Transfer Protocol (TFTP) 17
TimeFinder/FS
 integration considerations 5, 20
Trivial File Transfer Protocol (TFTP)
 protocol 17

U

Unicode characters 5
unmounting file systems from Data Mover
 permanent 55
 temporary 55

V

volume
 available space 59
 BCV configuration 13
 capacity, checking 59
 cloning 60
 concepts 7
 creation 24
 deleting 61, 62
 management, creating
 metavolume 27
 stripe volume 26, 27
 metavolume
 addressing 12
 configuration 12
 definition 4
 renaming 59
 size 59
 slice volume
 definition 4
 illustration 10
 stripe volume

configuration guidelines 19
performance 11

About this document

As part of its effort to continuously improve and enhance the performance and capabilities of the Celerra Network Server product line, EMC periodically releases new versions of Celerra hardware and software. Therefore, some functions described in this document may not be supported by all versions of Celerra software or hardware presently in use. For the most up-to-date information on product features, see your product release notes. If your Celerra system does not offer a function described in this document, contact your EMC Customer Support Representative for a hardware upgrade or software update.

Comments and suggestions about documentation

Your suggestions will help us improve the accuracy, organization, and overall quality of the user documentation. Send a message to techpubcomments@EMC.com with your opinions of this document.

Copyright © 1998-2008 EMC Corporation. All rights reserved.

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

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

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