



**EMC® VNX® Series**

**Release 7.1**

**Using VNX® Replicator**

**P/N 300-013-441 Rev 02**

**EMC Corporation**

*Corporate Headquarters:*  
Hopkinton, MA 01748-9103  
1-508-435-1000  
[www.EMC.com](http://www.EMC.com)

---

Copyright © 1998 - 2013 EMC Corporation. All rights reserved.

Published December 2013

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

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

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

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

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

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

Corporate Headquarters: Hopkinton, MA 01748-9103

<b>Preface</b> .....	<b>9</b>
<b>Chapter 1: Introduction</b> .....	<b>11</b>
System requirements.....	12
Restrictions.....	13
Cautions and warnings.....	16
User interface choices.....	17
Related information.....	17
<b>Chapter 2: Concepts</b> .....	<b>19</b>
VNX Replicator session behavior.....	20
Replication destination options.....	20
Replication source objects.....	21
Many-to-one replication configuration.....	26
One-to-many replication configurations.....	27
Cascade replication configurations.....	31
One-to-many and cascade replication configurations.....	34
Data Mover interconnect.....	34
How ongoing replication sessions work.....	40
How one-time file system copy sessions work.....	42
Updating the destination site with source changes.....	43
Stopping a replication session.....	44
Starting a replication session.....	45
Reversing a replication session.....	47
Switching over a replication session.....	47
Failing over a replication session.....	50
When to use failover, switchover, and reverse.....	52

Deleting a replication session.....	53
Performing an initial copy by using disk or tape.....	54
Planning considerations.....	57
<b>Chapter 3: Upgrading from previous versions.....</b>	<b>63</b>
Upgrade from a previous release.....	64
Enable VNX Replicator.....	64
<b>Chapter 4: Configuring communication between VNX systems.....</b>	<b>65</b>
Prerequisites.....	66
Set up communication on the source.....	66
Set up communication on the destination.....	67
Verify communication between VNX systems.....	68
<b>Chapter 5: Configuring communication between Data Movers.....</b>	<b>69</b>
Prerequisites.....	70
Verify interface network connectivity.....	70
Set up one side of an interconnect.....	74
Set up the peer side of the interconnect.....	76
Validate interconnect information.....	77
<b>Chapter 6: Configuring a file system replication session.....</b>	<b>79</b>
Prerequisites.....	80
Verify destination storage.....	81
Validate Data Mover communication.....	81
Create a file system replication session.....	82
(Optional) Verify file system replication.....	85
<b>Chapter 7: Configuring a VDM replication session.....</b>	<b>87</b>
Prerequisites.....	88
Verify the source VDM.....	88
Verify available destination storage.....	88
Validate Data Mover communication.....	89
Create VDM replication session.....	90
Replicate the file system mounted to the VDM.....	91
(Optional) Verify VDM replication.....	92
<b>Chapter 8: Configuring a one-time copy session.....</b>	<b>95</b>
Prerequisites.....	96

Verify the source file system.....	96
Verify destination file system or storage.....	97
Validate Data Mover communication.....	99
Create a file system copy session.....	99
(Optional) Monitor file system copy.....	102
<b>Chapter 9: Configuring advanced replications.....</b>	<b>105</b>
Configure a one-to-many replication.....	106
Configure a cascade replication.....	106
Configure a one-to-many replication with common base checkpoints.....	107
Configure a cascading replication with common base checkpoints.....	115
Configure the replication of a file system mounted on an Nested Mount File System mountpoint.....	124
<b>Chapter 10: Managing replication sessions.....</b>	<b>127</b>
Get information about replication sessions.....	128
Modify replication properties.....	131
Refresh the destination.....	132
Delete a replication session.....	133
Stop a replication session.....	134
Start a replication session.....	136
Start a failed over or switched over replication.....	138
Start a replication session that is involved in a one-to-many configuration.....	139
Reverse the direction of a replication session.....	140
Switch over a replication session.....	141
Fail over a replication session.....	144
<b>Chapter 11: Managing replication tasks.....</b>	<b>149</b>
Monitor replication tasks.....	150
Abort a replication task.....	152
Delete a replication task.....	156
<b>Chapter 12: Managing Data Mover interconnects.....</b>	<b>159</b>
View a list of Data Mover interconnects.....	160
View Data Mover interconnect information.....	160
Modify Data Mover interconnect properties.....	161
Change the interfaces associated with an interconnect .....	163
Pause a Data Mover interconnect.....	168

Resume a paused Data Mover interconnect.....	169
Validate a Data Mover interconnect.....	169
Delete a Data Mover interconnect.....	170
<b>Chapter 13: Managing the replication environment.....</b>	<b>171</b>
Monitor replication.....	172
View the passphrase.....	172
Change the passphrase.....	173
Extend the size of a replicated file system.....	174
Change the percentage of system space allotted to SavVols.....	174
Change replication parameters.....	175
Mount the source or destination file system on a different Data Mover.....	176
Rename a Data Mover that has existing interconnects.....	176
Change the mount status of source or destination object.....	177
Perform an initial copy by using the disk transport method.....	178
Change Control Station IP addresses.....	180
Perform an initial copy by using the tape transport method.....	183
Change the NDMP server_param value.....	185
Recover from a corrupted file system using nas_fsck.....	185
Manage expected outages.....	186
Manage unexpected outages.....	188
<b>Chapter 14: Troubleshooting.....</b>	<b>191</b>
EMC E-Lab Interoperability Navigator.....	192
Error messages.....	192
Log files.....	192
Return codes for nas_copy.....	193
Interconnect validation failure.....	195
Replication fails with quota exceeded error.....	195
Two active CIFS servers after failover or switchover.....	195
Network performance troubleshooting.....	196
EMC Training and Professional Services.....	197
<b>Appendix A: Setting up the CIFS replication environment.....</b>	<b>199</b>
Verify IP infrastructure.....	202
Configure an interface.....	204
Set up DNS.....	206
Synchronize Data Mover and Control Station Time.....	207
Synchronize Data Mover source site time.....	207

Synchronize Data Mover destination site time.....208

Synchronize Control Station source site time.....208

Synchronize Control Station destination site time.....210

Configure user mapping.....210

Prepare file systems for replication.....211

**Glossary.....213**

**Index.....221**





## Preface

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

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

---

## Special notice conventions

EMC uses the following conventions for special notices:

---

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

---

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

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

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

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

---

## Where to get help

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

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

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

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

---

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

---

---

## Your comments

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

Please send your opinion of this document to:

[techpubcomments@EMC.com](mailto:techpubcomments@EMC.com)

This document describes how to perform replication on EMC VNX by using the latest version of replication, EMC VNX Replicator. This version of replication enables you to create and manage replication sessions, each producing a read-only, point-in-time copy of a given source object at a designated destination. VNX Replicator sessions, which can be created by using the command line interface (CLI) or Unisphere, are characterized by an architecture based on snapshot/checkpoint technology, asynchronous transfer to the destination and support for file system and Virtual Data Mover (VDM) source object types.

This document is part of the VNX information set and is intended for use by system administrators who are responsible for establishing replication in the VNX environment. Before using VNX Replicator, system administrators establishing replication should understand NAS volumes and file systems.

This section discusses:

- ◆ [System requirements on page 12](#)
- ◆ [Restrictions on page 13](#)
- ◆ [Cautions and warnings on page 16](#)
- ◆ [User interface choices on page 17](#)
- ◆ [Related information on page 17](#)

## System requirements

This section details the EMC® VNX® Series software, hardware, network, storage, and network settings to use EMC VNX Replicator.

### Local or loopback replication

[Table 1 on page 12](#) describes the VNX requirements for local or loopback replication.

**Table 1. System requirements for local or loopback replication**

Software	<ul style="list-style-type: none"> <li>◆ VNX version 7.1 or later.</li> <li>◆ Licenses for VNX Replicator, EMC SnapSure™, and CIFS.</li> <li>◆ If an application-consistent copy of an iSCSI LUN or NFS shared file system is required, you should use Replication Manager version 5.0 or later with the necessary patches. The <i>Replication Manager Release Notes</i> provide the latest information.</li> </ul>
Hardware	<ul style="list-style-type: none"> <li>◆ One VNX for file-storage EMC Symmetrix® or EMC VNX for block pair.</li> </ul>
Network	<ul style="list-style-type: none"> <li>◆ IP addresses configured for the source and destination Data Movers.</li> <li>◆ Loopback replication always uses IP address 127.0.0.1.</li> </ul>
Storage	<ul style="list-style-type: none"> <li>◆ Sufficient storage space available for the source and destination file systems.</li> <li>◆ Sufficient SavVol space available for use.</li> </ul>

### Remote replication

[Table 2 on page 12](#) describes the VNX requirements for remote replication.

**Table 2. System requirements for remote replication**

Software	<ul style="list-style-type: none"> <li>◆ VNX version 7.1 or later.</li> <li>◆ Licenses for VNX Replicator, SnapSure, and CIFS.</li> </ul>
Hardware	<ul style="list-style-type: none"> <li>◆ Minimum of two VNX for File-storage Symmetrix or VNX for Block pairs.</li> </ul>

Network	<ul style="list-style-type: none"> <li>◆ IP addresses configured for the source and destination Data Movers (port 8888 used by replication for transferring data—contact EMC Customer Support to change this setting).</li> <li>◆ HTTPS connection between the source and destination Data Movers (port 5085—cannot be changed).</li> <li>◆ HTTPS connection between the Control Station on the source site and the Control Station on the destination site (port 443—cannot be changed).</li> </ul> <p style="text-align: center;">—————</p> <p><b>Note:</b> IP connection between the Control Station external IP address and the Data Mover external IP address is not required.</p> <p style="text-align: center;">—————</p> <ul style="list-style-type: none"> <li>◆ Internet Control Message Protocol (ICMP) is required. ICMP ensures that a destination VNX is accessible from a source VNX. The ICMP protocol reports errors and provides control data about IP packet processing.</li> </ul>
Storage	<ul style="list-style-type: none"> <li>◆ Sufficient storage space available for the source and destination file systems.</li> <li>◆ Sufficient SavVol space available for use.</li> </ul>
Security	<ul style="list-style-type: none"> <li>◆ The same VNX for File administrative user account with the nasadmin role must exist on both the source and destination VNX for File systems.</li> </ul>

## Restrictions

The following restrictions apply to VNX Replicator.

### General restrictions

- ◆ VNX Replicator requires a license.
- ◆ VNX Replicator is not supported with Network Address Translation (NAT).
- ◆ To take advantage of a one-to-many configuration (one source to many destinations), you must create multiple replication sessions on the source. Each session associates the source with up to four different destinations.
- ◆ For one-to-many configurations, EMC supports only one failed over replication session per source object. If you fail over more than one session for a given source object, you can only save the changes from one of the replication sessions.
- ◆ To take advantage of a cascade configuration (where one destination also serves as the source for another replication session), you must create a replication session on each source in the path. Cascade is supported for two network hops.
- ◆ The maximum number of active sessions is 256 per Data Mover. This limit includes all active replication sessions as well as copy sessions. Any additional sessions are queued until one of the current sessions completes.

- ◆ The maximum number of initial copies that can be in progress at one time is 16. This limit includes all active replication sessions as well as copy sessions.
- ◆ If you specify a name service interface name for an interconnect, the name must resolve to a single IP address, for example by a DNS server. However, it is not required to be a fully qualified name.
- ◆ After you configure an interconnect for a Data Mover, you cannot rename that Data Mover without deleting and reconfiguring the interconnect and reestablishing replication sessions.
- ◆ VNX Replicator works with disaster recovery replication products such as EMC SRDF®/Synchronous (SRDF/S) and SRDF/Asynchronous (SRDF/A) or EMC MirrorView™/Synchronous (MirrorView/S). You can run SRDF or MirrorView/S products and VNX Replicator on the same data. However, if there is an SRDF or MirrorView/S site failover, you cannot manage VNX Replicator sessions on the SRDF or MirrorView/S failover site. Existing VNX Replicator sessions will continue to run on the failed over Data Mover and data will still be replicated. On the primary site, you can continue to manage your SRDF or MirrorView/S replication sessions after the restore.
- ◆ If you plan to enable international character sets (Unicode) on your source and destination sites, you must first set up translation files on both sites before starting Unicode conversion on the source site. *Using International Character Sets on VNX for File* describes this action in detail.
- ◆ VNX Replicator is not supported in a network environment that has Cisco Wide Area Application Services (WAAS) setup. If you use remote replication in a network environment that has WAAS devices, the WAAS devices must be placed between Control Stations and configured in pass-through mode for all the ports used by replication. Otherwise, the devices may interfere with replication. When configured for pass-through communication, traffic will pass through the network unoptimized, but it will allow the WAAS device to be used for replication traffic between the remote sites.
- ◆ File system replication with user level checkpoints was introduced in Celerra Network Server version 6.0.41. It is not supported in VNX for file version 7.0, but is supported in VNX for file version 7.0.28.0 and later. When performing an upgrade from Celerra Network Server version 6.0.41, you must upgrade to VNX for file version 7.0.28.0 or later to continue replicating file systems by using user level checkpoints.

---

### File system restrictions

- ◆ A given source file system supports up to four replication sessions (one-time copy or ongoing replication sessions).
- ◆ For file systems with file-level retention (FLR):
  - You cannot create a replication session or copy session unless both the source and destination file systems have the same FLR type, either off, enterprise, or compliance. Also, when creating a file system replication, you cannot use an FLR-C-enabled destination file system that contains protected files.

- You cannot start a replication session if an FLR-C-enabled destination file system contains protected files and updates have been made to the destination.
- You cannot start a switched over replication session in the reverse direction when an FLR-C-enabled original source file system contains protected files and updates have been made to the original source since the switchover.
- ◆ File systems enabled for processing by VNX for File data deduplication may be replicated by using VNX Replicator as long as the source and destination file system formats are the same. For example, VNX for File can replicate deduplicated file systems between systems running version 5.6.47 or later of the Operating Environment.
- ◆ File-level deduplication (compression) consumes remote replication network bandwidth that can impact WAN-connected sites by causing larger data transmissions. File-level deduplication of a file causes the original data blocks to be freed, making them available for reallocation and reuse by other files.

A copy-on-write replication operation is triggered during:

- Reuse of the freed data blocks.
- The file compression process which requires allocation and writing of the data blocks to store the compressed data.

If a network bandwidth that is allocated for geographically separate locations with a WAN is extremely limited, significant delays in transmission of the delta sets can occur while compression is performed on a file system. Both file-level deduplication and FileMover create a delta set equal to the number of blocks in the original file; the largest number of files is processed on the first run.

- ◆ FileMover consumes remote replication network bandwidth that can impact WAN-connected sites by causing larger data transmissions. File-level archiving of individual files causes the original data blocks to be freed, making them available for reallocation and reuse by other files. This operation triggers a copy-on-write replication of blocks. If a network bandwidth that is allocated for geographically separate locations with a WAN is extremely limited, significant delays in transmission of the delta sets can occur while archiving is performed on a file system. Both file-level deduplication and FileMover create a delta set equal to the number of blocks in the original file; the largest number of files is processed on the first run.
- ◆ You cannot unmount a file system if replication is configured. Replication has to be stopped for an unmount of the file system to succeed. When the file system is unmounted, the internal checkpoints will be automatically unmounted and when mounted, the internal checkpoints will be automatically mounted.
- ◆ You cannot move a file system to another Data Mover if replication is configured on the file system. You must stop replication before the file system is unmounted. After the file system is mounted on the new Data Mover, start the replication session.
- ◆ VNX File System Migration is unsupported (an mgfs file system cannot be replicated).

- ◆ EMC VNX Multi-Path File System (MPFS) is supported on the source file system, but not on the destination file system.
- ◆ For EMC TimeFinder® /FS:
  - A file system cannot be used as a TimeFinder destination file system and as a destination file system for replication at the same time.
  - Do not use the TimeFinder/FS -restore option for a file system involved in replication.
- ◆ When replicating databases, additional application-specific considerations may be necessary to bring the database to a consistent state (for example, quiescing the database).
- ◆ Replication failover is not supported for CIFS local users and local groups unless you use Virtual Data Movers (VDMs). *Configuring Virtual Data Movers on VNX* describes VDM configuration.
- ◆ Modifying properties of file systems that are replicated from legacy systems or VNX systems to VNXe systems can result in the replication sessions not functioning as expected. This is because the feature set for VNXe is limited when compared to the feature set of VNX. Refer to the feature documentation to ensure that it is supported on all products in the replication relationship.

---

### VDM replication restrictions

- ◆ A given VDM supports up to four replication sessions.
- ◆ Replication of a physical Data Mover root file system is not supported. VDM root file systems are supported.
- ◆ Single domains on a VDM are supported. Multiple domains without a trust relationship are not supported on a VDM. This restriction also applies to a physical Data Mover.
- ◆ Do not load a VDM on the source and destination sites simultaneously. This ensures that the Active Directory and DNS do not contain conflicting information for CIFS server name resolution.
- ◆ A VDM replication session cannot be switched over, failed over, or reversed if the internationalization settings of the source and destination Data Movers do not match.
- ◆ The VDM for NFS solution allows configuration of an NFS server, also known as an NFS endpoint, per VDM. The NFS endpoint of a replicated VDM works only if the Operating Environments of the source and destination sites are version 7.0.50.0 or later.

---

### Cautions and warnings

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



- ◆ Replicating file systems from a Unicode-enabled Data Mover to an ASCII-enabled Data Mover is not supported. I18N mode (Unicode or ASCII) must be the same on both the source and destination Data Movers.
- ◆ VNX provides you with the ability to fail over a CIFS server and its associated file systems to a remote location. In a disaster, because VNX Replicator is an asynchronous solution, there might be some data loss. However, the file systems will be consistent (fsck is not needed). When using databases, additional application-specific considerations might be necessary to bring the database to a consistent state (for example, quiescing the database).

---

## User interface choices

VNX offers flexibility in managing networked storage based on the support environment and interface preferences. This document describes how to set up and manage VNX Replicator by using the CLI. With the VNX Replicator license enabled, you can also perform VNX Replicator tasks by using EMC Unisphere™, available by selecting Data Protection > Mirrors and Replication > Replications for File.

To enable the VNX Replicator license in Unisphere:

1. Select **Settings**.
2. Under **More Settings**, click **Manage Licenses for File**.
3. Select **ReplicatorV2 Licensed** from the options.
4. Click **Apply**.
5. Click **OK** to confirm that you have a license for this feature and you want to enable its use.

After the license is enabled, you have to restart the management application to see this change take effect in the user interface.

For additional information about managing your VNX:

- ◆ Unisphere online help
- ◆ The *EMC VNX Operating Environment for File Release Notes* contain additional, late-breaking information about VNX management applications.

---

## Related information

Specific information related to the features and functionality described in this document are included in the following documents:

- ◆ *Using VNX SnapSure*
- ◆ *VNX Command Line Interface Reference for File*
- ◆ Online VNX man pages

- ◆ *Parameters Guide for VNX for File*
- ◆ *Configuring and Managing CIFS on VNX*
- ◆ *Configuring Virtual Data Movers on VNX*
- ◆ *VNX Glossary*
- ◆ *EMC VNX Operating Environment for File Release Notes*
- ◆ *Configuring Events and Notifications on VNX for File*
- ◆ *VNX System Operations*
- ◆ *Using VNX File-Level Retention*
- ◆ *Managing Volumes and File Systems for VNX Manually*
- ◆ *Managing Volumes and File Systems with VNX Automatic Volume Management*
- ◆ *Celerra Network Server Error Messages Guide*
- ◆ *Configuring and Managing Networking on VNX*
- ◆ *Configuring and Managing Network High Availability on VNX*

---

### **EMC VNX documentation on EMC Online Support**

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

---

### **VNX wizards**

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

The concepts and planning considerations to understand VNX Replicator are:

- ◆ [VNX Replicator session behavior on page 20](#)
- ◆ [Replication destination options on page 20](#)
- ◆ [Replication source objects on page 21](#)
- ◆ [Many-to-one replication configuration on page 26](#)
- ◆ [One-to-many replication configurations on page 27](#)
- ◆ [Cascade replication configurations on page 31](#)
- ◆ [One-to-many and cascade replication configurations on page 34](#)
- ◆ [Data Mover interconnect on page 34](#)
- ◆ [How ongoing replication sessions work on page 40](#)
- ◆ [How one-time file system copy sessions work on page 42](#)
- ◆ [Updating the destination site with source changes on page 43](#)
- ◆ [Stopping a replication session on page 44](#)
- ◆ [Starting a replication session on page 45](#)
- ◆ [Reversing a replication session on page 47](#)
- ◆ [Switching over a replication session on page 47](#)
- ◆ [Failing over a replication session on page 50](#)
- ◆ [When to use failover, switchover, and reverse on page 52](#)
- ◆ [Deleting a replication session on page 53](#)
- ◆ [Performing an initial copy by using disk or tape on page 54](#)
- ◆ [Planning considerations on page 57](#)

---

## VNX Replicator session behavior

The behavior for replication sessions and basic copy sessions can be summarized as follows:

- ◆ The destination for any replication or basic copy session can be the local VNX (same Data Mover or different Data Mover) or a remote VNX. Any session with a remote destination requires a trusted communication path between the local and remote VNX with a common passphrase. If no established connection exists between the local and remote VNXs, you must first create it. [Chapter 4](#) describes this procedure.
- ◆ Any replication or basic copy session requires a configured connection between the source object's Data Mover and the destination Data Mover. This Data Mover-to-Data Mover connection is called an interconnect.

---

**Note:** The interconnect between Data Movers is Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) in anonymous mode. This HTTPS setting indicates a secure HTTP connection and cannot be configured by users

---

If no established interconnect exists between the source and destination Data Mover, you must first create it. When you create an interconnect (one per Data Mover pair), you list the IP addresses that are available to sessions by using the interconnect, and optionally, a bandwidth schedule that lets you control the bandwidth that is available to sessions during certain time periods. You create an interconnect between a given source and destination Data Mover pair on each VNX. [Chapter 5](#) describes this procedure.

- ◆ When you create a replication session, it automatically creates two internal checkpoints on the source and destination, for example, file system checkpoints. Using these checkpoints, the replication session copies the changes found in the source object to the destination object. [How ongoing replication sessions work on page 40](#) provides more details.
- ◆ Each replication session to a designated destination produces a point-in-time copy of the source object, and the transfer is performed asynchronously.
- ◆ For each replication session, you can specify an update policy in which you specify a maximum amount of time that the source and destination can be out of synchronization before an update is automatically performed, or you can update the destination at will by issuing a refresh request for the session.
- ◆ You can create either multiple replication sessions or basic copy sessions that copy the same source object to multiple destinations. This is called a one-to-many configuration.
- ◆ You can set up a cascading replication configuration where a destination site can serve as a source for another replication session to another destination, up to two network hops.

---

## Replication destination options

There are three types of replication sessions:

- ◆ Loopback replication
- ◆ Local replication
- ◆ Remote replication

---

### Loopback replication

Replication of a source object occurs on the same Data Mover in the cabinet. Communication is established by using a predefined Data Mover interconnect established automatically for each Data Mover in the cabinet.

---

### Local replication

Replication occurs between two Data Movers in the same VNX for file cabinet. Both Data Movers must be configured to communicate with one another by using a Data Mover interconnect. After communication is established, a local replication session can be set up to produce a read-only copy of the source object for use by a Data Mover in the same VNX for file cabinet. For file system replication, the source and destination file systems are stored on separate volumes.

---

### Remote replication

Replication occurs between a local Data Mover and a Data Mover on a remote VNX system. Both VNX for file cabinets must be configured to communicate with one another by using a common passphrase, and both Data Movers must be configured to communicate with one another by using a Data Mover interconnect. After communication is established, a remote replication session can be set up to create and periodically update a source object at a remote destination site. The initial copy of the source file system can either be done over an IP network or by using the tape transport method.

After the initial copy, replication transfers changes made to the local source object to a remote destination object over the IP network. These transfers are automatic and are based on definable replication session properties and update policy.

---

## Replication source objects

VNX Replicator can replicate the following types of source objects:

- ◆ File systems (ongoing)
- ◆ File systems (one-time copy)
- ◆ VDMs

---

## File system replication

File system replication creates a read-only, point-in-time copy of a source production file system at a destination and periodically updates this copy, making it consistent with the source file system. You can use this type of replication for content distribution, backup, reporting, and software testing.

You can create up to four replication sessions for a particular source file system. Therefore, one-to-many and cascade configurations are supported.

### Replication and deduplication

Deduplicating the contents of a file system before it is replicated by using VNX Replicator can greatly reduce the amount of data that has to be sent over the network as part of the initial baseline copy process. After replication and deduplication are running together, the impact of deduplication on the amount of data transferred over the network will depend on the relative timing of replication updates and deduplication runs. In all but the most extreme circumstances, replication updates will be more frequent than deduplication scans of a file system. This means that new and changed data in the file system will almost always be replicated in its nondeduplicated form first, and any subsequent deduplication of that data will prompt additional replication traffic. This effect will be true of any deduplication solution that post-processes data and updates remote replicas of the data set more frequently than the deduplication process is run.

Deduplication can add a maximum of 25 MB/s of write activity during processing. If you are using a lower bandwidth network, you should check to ensure that your replication bandwidth can accommodate the additional load and that your SLAs are being met.

[How ongoing replication sessions work on page 40](#) provides an example of how replication works by using file system as an example. [Chapter 6](#) describes how to create a file system replication session.

---

## One-time file system copy

Use file system one-time copy to perform a full copy of a file system mounted read-only, a file system writeable checkpoint mounted read-only, or a file system checkpoint. A file system checkpoint is by default a differential copy if an earlier checkpoint that serves as a common base exists; otherwise, it is a full copy. You perform a file system copy with the `nas_copy` command.

During a copy session, the destination file system is mounted read-only on the Data Mover. The read activity is redirected to the checkpoint at the start of the copy session.

Multiple copy sessions can run from the same source to multiple destinations, so a one-to-many configuration is allowed.

[How one-time file system copy sessions work on page 42](#) explains how the copy process works. [Chapter 8](#) describes how to create a copy session.

## VDM replication

VDM replication supports only CIFS and NFS protocols. It accommodates the CIFS working environment and replicates information contained in the root file system of a VDM. This form of replication produces a point-in-time copy of a VDM that re-creates the CIFS environment at a destination. However, it does not replicate the file systems mounted to a VDM.

In CIFS environments, for a CIFS object to be fully functional and accessible on a remote VNX, you must copy the complete CIFS working environment, which includes local groups, user mapping information, Kerberos objects, shares, event logs, and registry information to the destination site. This CIFS working environment information is in the root file system of the VDM.

**Note:** The Internal Usermapper data is not stored in the VDM.

In addition to the environmental information, you must replicate the file systems associated with the CIFS server.

VDMs provide virtual partitioning of the physical resources and independently contain all the information necessary to support the contained servers. Having the file systems and the configuration information in one manageable container ensures that the data and the configuration information that make the data usable can fail over together. [Figure 1 on page 23](#) shows this concept.

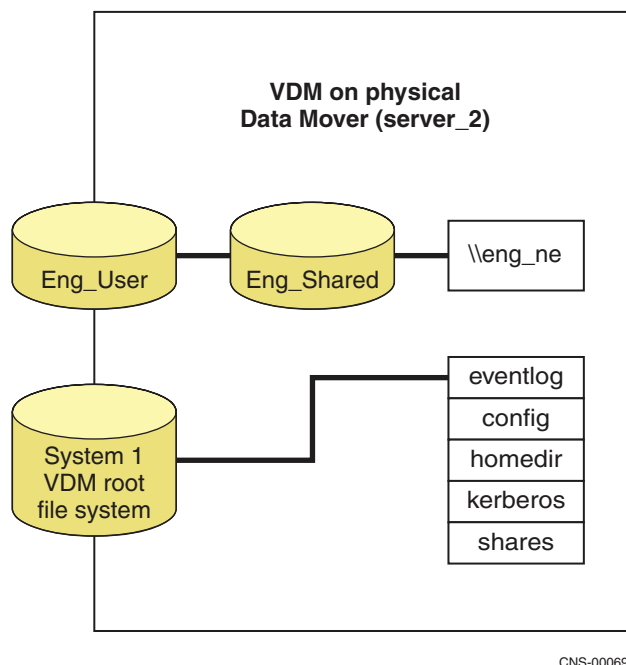


Figure 1. VDM container and concepts

You replicate the VDM first, then replicate the file systems mounted to a VDM. Replication provides the data transport and a VDM stores the relevant CIFS configuration information.

Source VDMs can be involved in replicating up to four destination VDMs concurrently. A destination VDM must be the same size as the source and in the mounted read-only state. A destination VDM can be a source for up to three other replications. Since a given VDM can act as both a source and destination for multiple replications, it supports a one-to-many and cascade configuration.

[Appendix A](#) provides information you should read before you create a VDM session. [Chapter 7](#) describes how to create a VDM replication session.

The VDM for NFS multi-naming domain solution for the Data Mover in the UNIX environment is implemented by configuring an NFS server, also known as an NFS endpoint, per VDM. The VDM is used as a container for information including the file systems that are exported by the NFS endpoint. The NFS exports of the VDM are visible through a subset of the Data Mover network interfaces assigned for the VDM. The same network interface can be shared by both CIFS and NFS protocols. However, only one NFS endpoint and CIFS server is addressed through a particular logical network interface.

The exports configuration for this NFS end-point is stored in the `vdm.cfg` file located in the VDM root file system. The VDM root file system is replicated as part of a VDM replication session.

The VDM for NFS multi-naming domain solution is not supported by versions of the Operating Environment earlier than 7.0.50.0. Therefore, when you replicate a VDM from a source system that runs on version 7.0.50.0 and later of the VNX Operating Environment to a destination system that runs on any version of the Operating Environment earlier than 7.0.50.0, the file systems replicate correctly, but the NFS endpoint does not work on the system with the Operating Environment earlier than version 7.0.50.0.

---

### Source objects and destination options

[Table 3 on page 25](#) provides a summary of the type of source objects that can be replicated, their corresponding destination options, and a brief description of when to use the object.



Table 3. Summary of source objects and destination options

Source object	Destination options	Description
File system, for ongoing replication (existing file system mounted read/write)	<ul style="list-style-type: none"> <li>◆ Same Data Mover</li> <li>◆ Different Data Mover in the same VNX cabinet</li> <li>◆ Data Mover in a VNX cabinet at a remote site, which protects against a site failure</li> <li>◆ Multiple destinations each using a different session, up to four</li> <li>◆ One destination serving as source for another replication session (cascade to two levels)</li> </ul>	<p>Use file system replication to produce a read-only, point-in-time copy of a source production file system at a destination and periodically update this copy, making it consistent with the source file system. Use this type of replication for content distribution, backup, and application testing.</p> <p>You can create up to four replication sessions for a given source file system.</p>
File system, for one-time copy (existing file system mounted read-only, writeable file system checkpoint mounted read-only, or existing file system checkpoint)	<ul style="list-style-type: none"> <li>◆ Same Data Mover</li> <li>◆ Different Data Mover in the same VNX cabinet</li> <li>◆ Data Mover in a VNX cabinet at a remote site</li> <li>◆ Multiple destinations each using a different session</li> </ul>	<p>Use file system one-time copy to perform a full copy of a file system mounted read-only, a file system writeable checkpoint mounted read-only, or a copy of a checkpoint file system. A file system checkpoint is by default a differential copy if an earlier checkpoint that serves as a common base exists. Otherwise, it is a full copy. You perform a file system copy with the <code>nas_copy</code> command.</p>
VDM in loaded state	<ul style="list-style-type: none"> <li>◆ Same Data Mover</li> <li>◆ Different Data Mover in the same VNX cabinet</li> <li>◆ Data Mover in a VNX cabinet at a remote site, which protects against site failure</li> <li>◆ Multiple destinations each using a different session, up to four</li> <li>◆ One destination serving as source for other replication sessions (cascade to two levels)</li> </ul>	<p>Use VDM replication to accommodate the CIFS working environment and replicate information contained in the root file system of a VDM. This form of replication produces a point-in-time copy of a VDM that re-creates the CIFS environment at a destination. However, it does not replicate the file systems mounted to a VDM. You replicate the VDM first, then replicate the file systems mounted to a VDM. Replication provides the data transport and a VDM stores the relevant CIFS configuration information.</p> <p>You can create up to four replication sessions for a given source VDM.</p>

---

## Many-to-one replication configuration

A many-to-one configuration, also known as edge-to-core replication, is when multiple VNX systems replicate to one centralized VNX. This configuration is equivalent to having multiple one-to-one replication sessions from multiple remote source VNX systems that share a common target destination VNX storage platform. This configuration is typically used to support remote consolidated backups.

In this configuration you are not replicating multiple file systems to a single file system. Rather, you are replicating distinct file systems (fs1, fs2, and fs3), which may or may not exist on the same VNX, to distinct file systems that reside on a common target destination VNX.

The number of inbound replications supported on the target VNX is limited only by the file system and replication session limits.

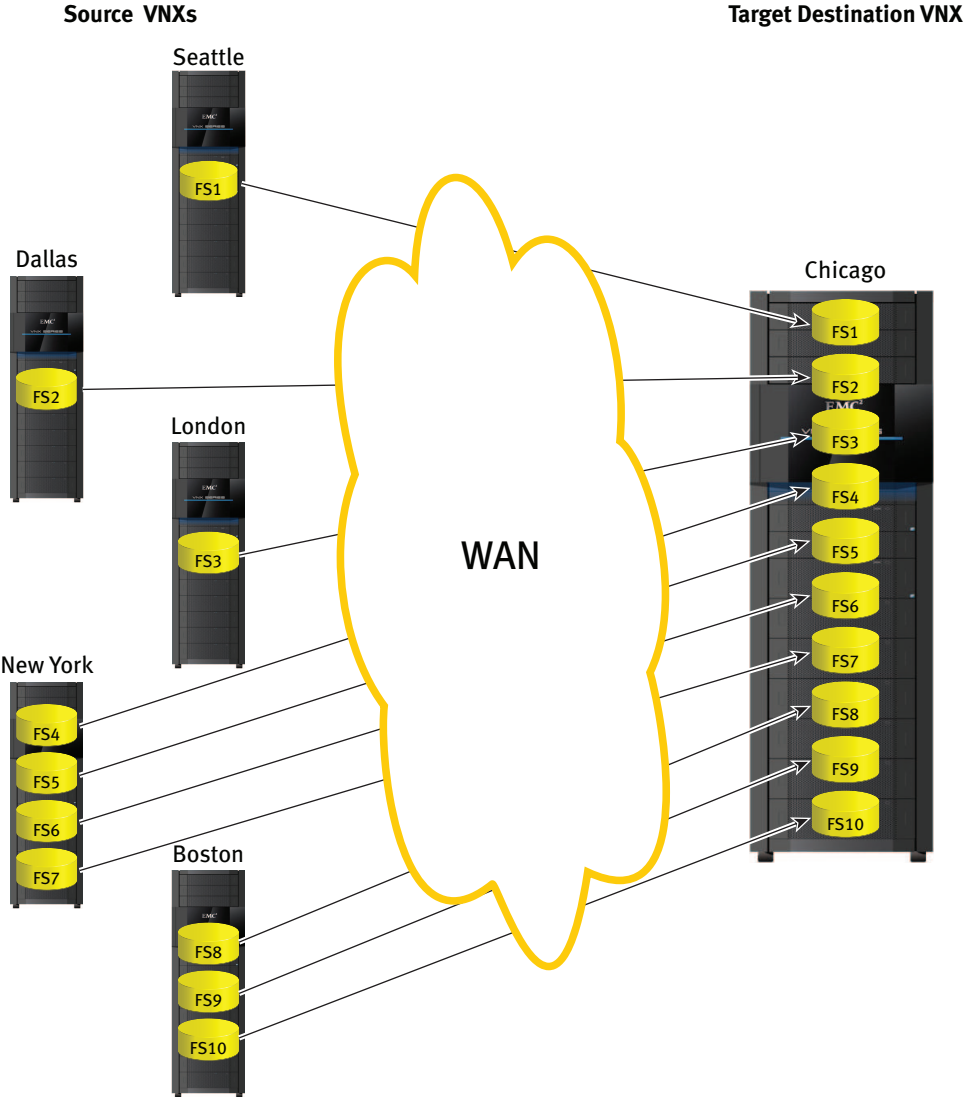
---

**Note:** [Restrictions on page 13](#) describes specific replication session and file system limitations. Also, the EMC E-Lab™ Interoperability Navigator provides access to the EMC interoperability support matrices.

---

Figure 2 on page 27 shows a basic many-to-one replication configuration for ongoing replication of 10 different file systems from 5 remote source VNX systems to one common target destination VNX located in Chicago.

Figure 2. Many-to-one replication configuration



Each replication session shares the same target destination VNX storage platform

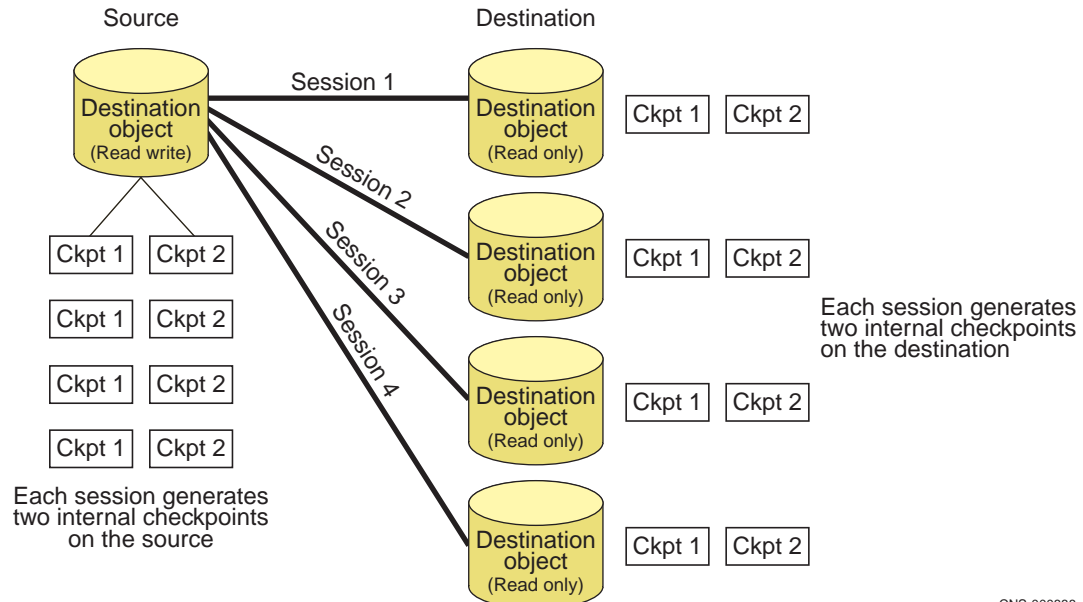
VNX-000046

### One-to-many replication configurations

A one-to-many configuration is where a source object is used in one or multiple replication sessions. This configuration requires a separate replication session for each destination.

Figure 3 on page 28 shows a basic one-to-many configuration for ongoing replication of a given source file system to up to four multiple destinations.

One to many configuration



CNS-000888

Figure 3. One-to-many file system replication

Configure a one-to-many replication on page 106 provides details on how to configure this environment.

You can create the maximum number of remote data copies (16) by using a one-to-many configuration to four locations and then cascade from each location to three additional sites:

- Source --> 1 --> 5, 6, 7
- Source --> 2 --> 8, 9, 10
- Source --> 3 --> 11, 12, 13
- Source --> 4 --> 14, 15, 16

You can create the maximum number of remote data copies (12) with a copy session reserved at the source site by using a one-to-many configuration to three locations and then cascade from each location to three additional sites:

- Source --> 1 --> 5, 6, 7
- Source --> 2 --> 8, 9, 10
- Source --> 3 --> 11, 12, 13
- Source --> 4 (reserved for copy)

If you reverse a replication session that is involved in a one-to-many configuration, the source side goes into a cascading mode. The original destination side from one of the sessions

becomes the source and the original source side becomes the destination and that source cascades out to the other destination sessions.

[Reversing a replication session on page 47](#) explains how this works in a one-to-many configuration.

[Switching over a replication session on page 47](#) and [Failing over a replication session on page 50](#) explain how replication works with these operations in a one-to-many configuration.

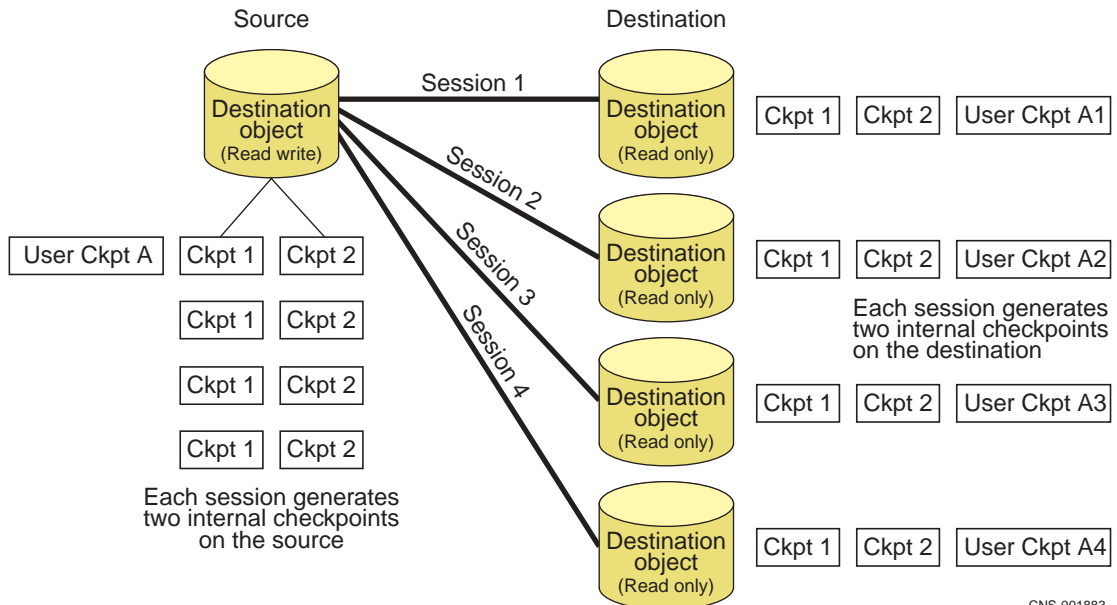
When you delete replications, the internal checkpoints are also deleted. To create future replications for any two file systems from the original configuration, you have to perform a full data copy. To avoid this scenario, create one or more user specified checkpoints on the source and destination file systems and refresh their replication by using these user checkpoints. This process transfers data from the user checkpoint of the source file system to the user checkpoint of the destination file system through the existing replication session. Consequently, the destination user checkpoint has the same view of the file system as the source user checkpoint. Such user checkpoints on the source and destination file systems, which have the same content are called common base checkpoints. This process is repeated for each destination in the one-to-many configuration.

After this process, even if you delete the replication sessions, you can configure new replication sessions between any two destinations because replication automatically identifies the common base checkpoints and avoids full data copy. You can create user checkpoints for file system and VDM replications. For VDM replications, you can create the user checkpoints on the root file system of the VDM. You cannot delete user checkpoints when they are being used by replication either for transfer or as an active common base.

[Figure 4 on page 30](#) shows a one-to-many configuration for file system replication with the optional user checkpoint and that you can replicate it to the other locations through the existing replication sessions. After the user specified checkpoints become the common bases for each destination location, if you delete any session, you can still configure replication from one file system to another through their common bases.

For example, if you delete Session 2, you can configure new replication sessions between Destination 1 and Destination 2 by using user checkpoints A1 and A2 because replication automatically identifies the common base checkpoints and avoids full data copy.

One to many configuration



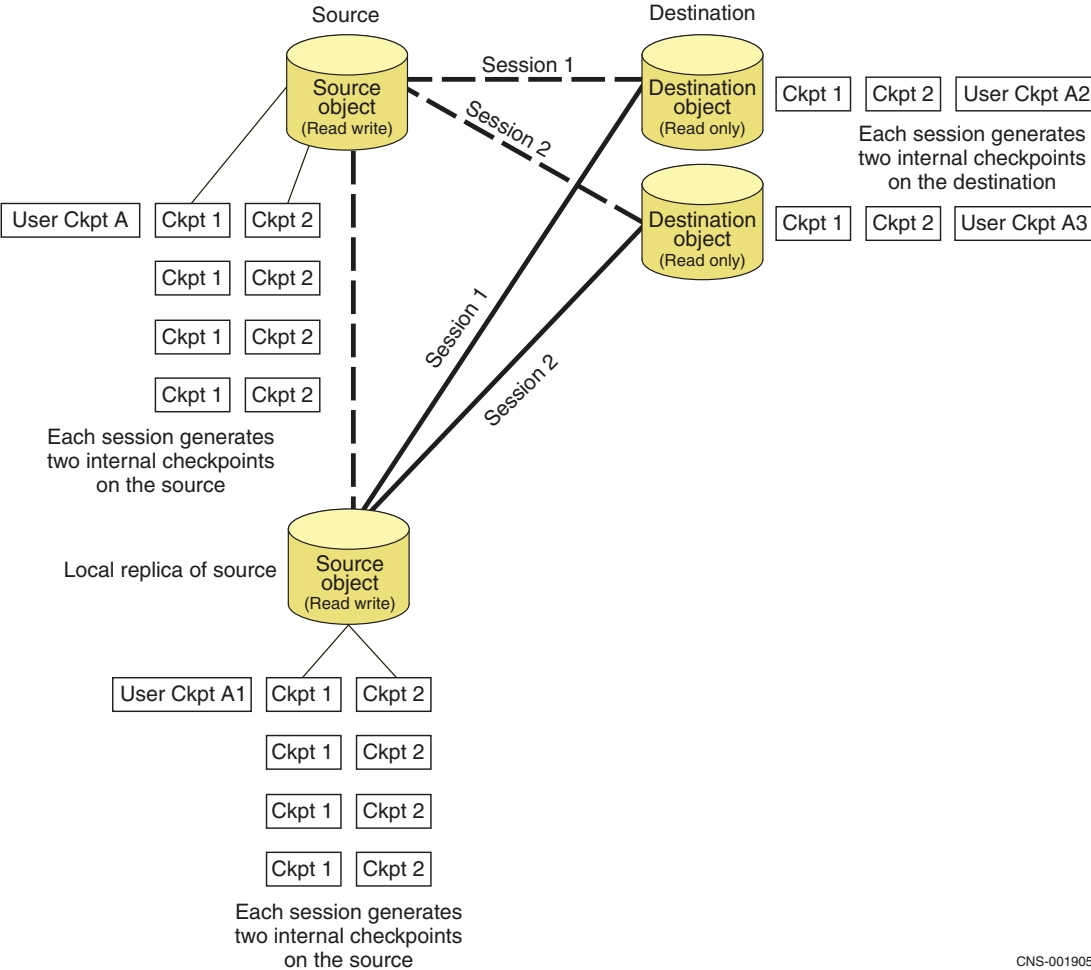
**Figure 4. One-to-many file system replication with user checkpoints**

When you replace source or destination file systems, the replications and internal checkpoints are deleted. In such a scenario, you will have to perform a full data copy to create future replications with the original configurations. To avoid performing a time-consuming full data copy over the WAN, create a local replica of the file system that you want to replace. You can then create one or more user specified checkpoints on the file system to be replaced, the local replica, and each of the other file systems involved in the replication.

After creating user specified checkpoints, you must refresh each replication by specifying the user checkpoint on the source and destination file system for that replication. Consequently, the user checkpoint of each destination file system (including local replica) has the same view of the file system as the user checkpoint on the source file system that

will be replaced. You can now use the local replica for replications without performing a full data copy over the WAN.

One to many configuration



CNS-001905

Figure 5. One-to-many file system replication to avoid full data copy over the WAN

Figure 5 on page 31 shows a one-to-many configuration for file system replication in the event that you want to replace the source file system and avoid a full data copy over the WAN.

### Cascade replication configurations

A cascade configuration is where one destination serves as the source for another replication session. Cascade is supported for two network hops. That is, a destination site can serve as a source for another replication session to another destination, but the cascade ends

there. You cannot configure that second destination as a source for yet another replication session. All source object types can be used in a cascade configuration.

**How it works**

A cascade configuration involves two replication sessions:

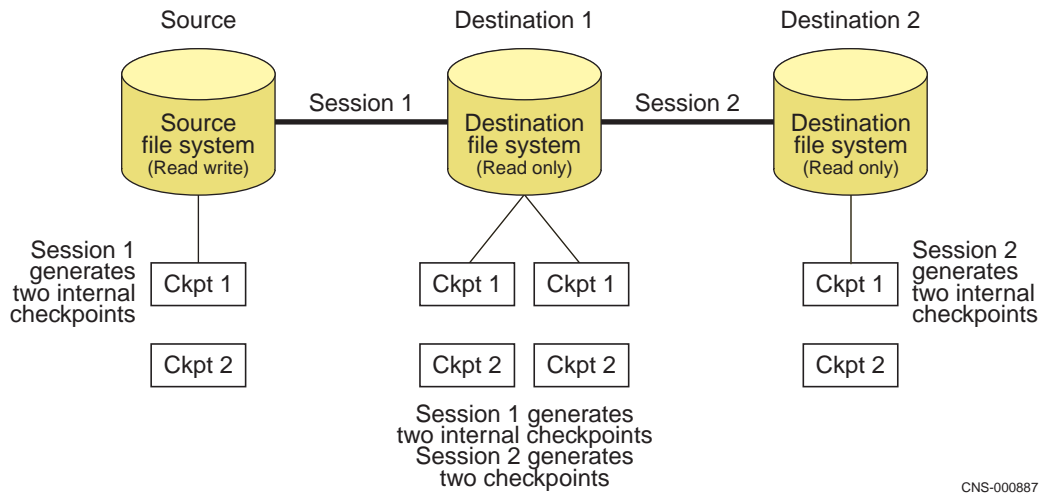
- ◆ Session 1 runs replication from Source to Destination 1.
- ◆ Session 2 runs replication from Destination 1 (serving as the source) to Destination 2.

When you set up a cascade configuration:

1. Create a replication session between Source and Destination 1.
2. Create a second replication session between Destination 1 and Destination 2.

Figure 6 on page 32 shows a basic cascade configuration for a file system replication. In this cascade configuration, Destination 1 serves as the source for replication Session 2.

Cascade replication



**Figure 6. One-level cascade configuration for file system replication**

Configure a cascade replication on page 106 provides details on how to configure this environment.

Figure 7 on page 33 shows a cascade configuration for file system replication with the optional user checkpoint, and how it is replicated from the source to Destination 1 and Destination 2 through replication Session 1 and replication Session 2, respectively. This enables the user checkpoints between Source, Destination 1, and Destination 2 to be



common bases for future replication. For a cascading replication, the replication must be refreshed for this first hop and then for the second hop.

Cascade replication

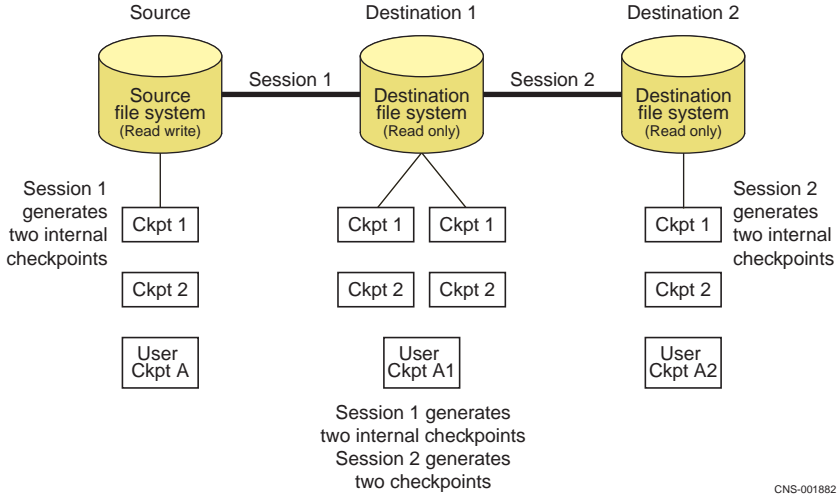
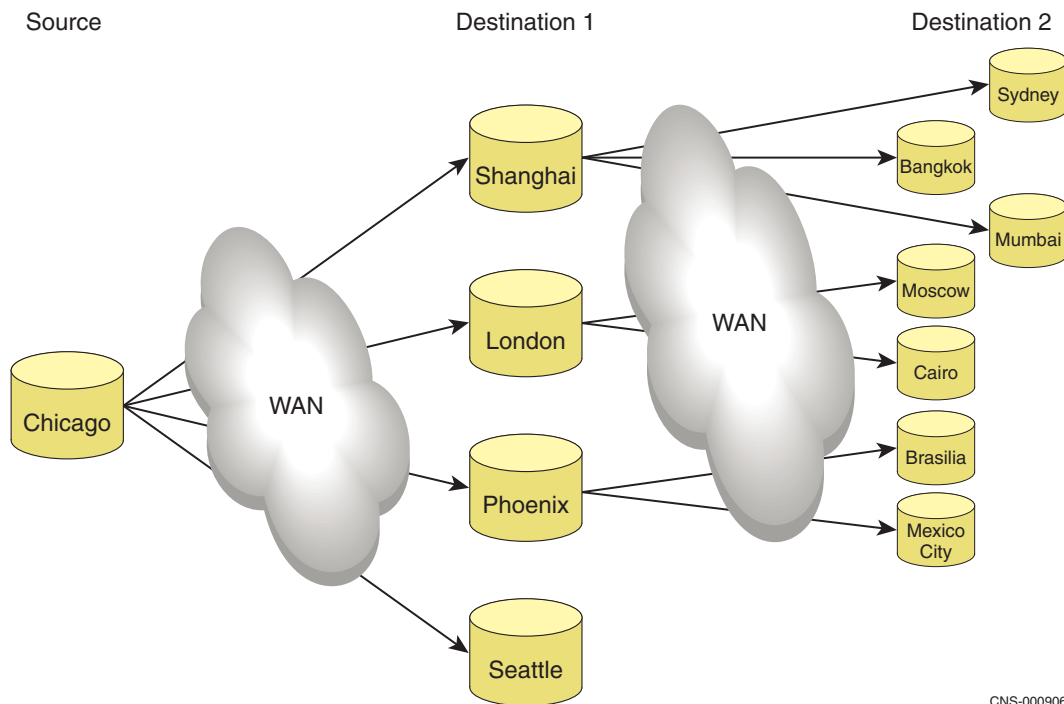


Figure 7. One-level cascade configuration for file system replication with user checkpoints

## One-to-many and cascade replication configurations

You can leverage one-to-many and cascade replication configurations to yield as many as 16 point-in-time copies—one source, up to four sessions and cascading to three more sessions. [Figure 8 on page 34](#) illustrates this concept.



CNS-000906

Figure 8. Combining a one-to-many and cascade replication configuration

## Data Mover interconnect

Each VNX Replicator session must use an established Data Mover interconnect, which defines the communication path between a given Data Mover pair located in the same cabinet or different cabinets. You define only one interconnect between a given Data Mover pair on a given VNX.

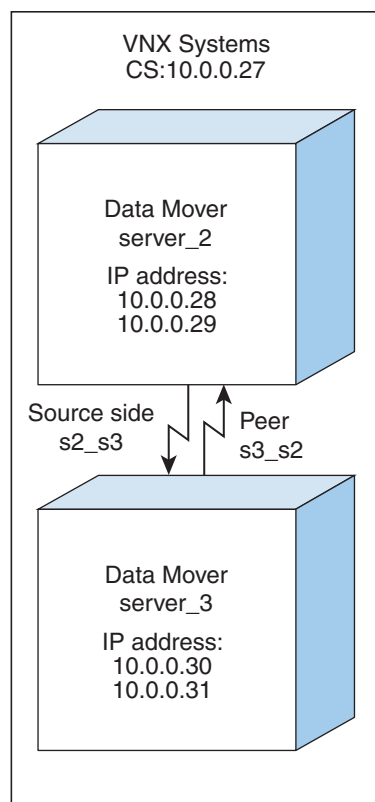
After you configure an interconnect for a Data Mover, you cannot rename that Data Mover without deleting and reconfiguring the interconnect and reestablishing replication sessions. [Chapter 13](#) provides more information about how to rename a Data Mover that has existing interconnects.

Both sides of a Data Mover interconnect must be established in each direction before you can create a replication session. This ensures communication between the Data Mover pair representing your source and destination.

**Note:** VNX Replicator uses a Cyclic Redundancy Check (CRC) method to perform additional error checking on the data sent over a Data Mover interconnect to ensure integrity and consistency. CRC is enabled automatically and cannot be disabled.

When you create the replication session, you specify the local side of an interconnect on which the source replication object resides, and this is the interconnect name displayed for the replication session. If the replication direction changes for a loopback or local session (for example, after a reverse), the direction of the arrow displayed for the Data Mover interconnect changes to indicate the new direction.

Figure 9 on page 35 shows an interconnect for a local replication.



VNX-000049

**Figure 9. Interconnect for a local replication**

### Types of interconnects

VNX Replicator supports three types of interconnects:

- ◆ Loopback interconnects — A loopback interconnect is automatically created for each Data Mover in the system. You cannot create, modify, or delete loopback interconnects. You simply select the loopback interconnect when setting up the replication session.

- ◆ Interconnects for local replication — These interconnects are created between a pair of Data Movers on the same system. You must create both sides of the interconnect on the local system before you can successfully create a local replication session.

When defining interconnects for a local replication, make the name meaningful by identifying the Data Movers (servers), for example:

s2\_s3 (local, or source side on the local system)

s3\_s2 (peer side on the same system)

- ◆ Interconnects for remote replication — These interconnects are created between a local Data Mover and a peer Data Mover on another system. You must create both sides of the interconnect, first on the local side and then on the peer side, before you can successfully create a remote replication session.

When defining interconnects for remote replication, make the name meaningful by identifying servers and VNX names or sites, for example:

s2CelA\_s3CelB (local side on the local system)

s3CelB\_s2CelA (peer side on the remote system)

or

NYs3\_LAs4 (local side on the local system)

LAs4\_NYs3 (peer side on the remote system)

#### Interconnect naming conventions

[Figure 10 on page 37](#) shows an example of the local and remote interconnects defined for a remote replication. In this sample configuration, the naming convention is based on the VNX series model number followed by the Data Mover server number.

The local interconnects defined for site A are:

- ◆ VNX3\_s2
- ◆ VNX3\_s3

The local interconnects defined for site B are:

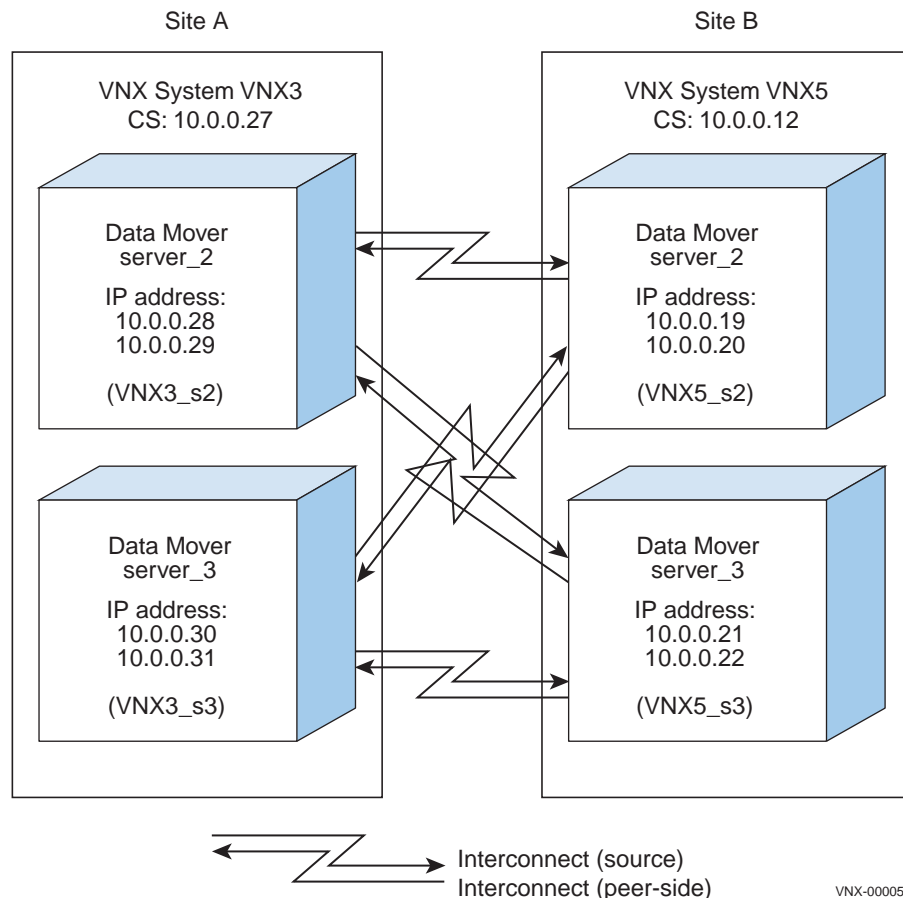
- ◆ VNX5\_s2
- ◆ VNX5\_s3

The remote interconnects defined for site A are:

- ◆ VNX3\_s2-VNX5\_s2
- ◆ VNX3\_s2-VNX5\_s3
- ◆ VNX3\_s3-VNX5\_s2
- ◆ VNX3\_s3-VNX5\_s3

The remote interconnects defined for site B are:

- ◆ VNX5\_s2-VNX3\_s2
- ◆ VNX5\_s2-VNX3\_s3
- ◆ VNX5\_s3-VNX3\_s2
- ◆ VNX5\_s3-VNX3\_s3



**Figure 10. Interconnect for a remote replication**

### Interconnect configuration elements

For each side of an interconnect, you specify the following elements:

- ◆ Source and destination interfaces — A list of source and destination network interfaces available to replication sessions. You define the interface list by using IP addresses IPv4 or IPv6) or name service interface names or a combination of both. But how you specify an interface determines whether a replication session later specifies the interface by interface name or IP address.

If you define an interface using an IP address, ensure the corresponding destination interface list uses the same IPv4/IPv6 network protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol (IPv4/IPv6). In addition, for each network protocol type specified in the source interface list, at least one interface from the same type must be specified in the destination list and vice versa. For example, if the source interface list includes one or more IPv6 addresses, the destination interface list must also include at least one IPv6 address.

The name service interface name is a fully qualified name given to a network interface. Name service interface names are not the same as VNX network device names (for example, cge0). Name service interface names are names that are resolved through a naming service (local host file, DNS, NIS, or LDAP).

- ◆ **Bandwidth schedule** — Allocates the interconnect bandwidth used on specific days, or specific times, or both, instead of by using all available bandwidth at all times. A bandwidth schedule applies to all replication sessions by using the interconnect. By default, an interconnect provides all available bandwidth at all times for the interconnect. [Specifying bandwidth usage for a Data Mover interconnect on page 38](#) provides more explanation and an example.

You can modify the name of an interconnect and the source and destination Data Mover interfaces as long as the interconnect is not in use by a replication session. To modify the peer side of an interconnect configured on a remote system, you must modify it from that system. You can also pause and resume data transmission over an interconnect.

[Planning considerations on page 57](#) provides interconnect setup considerations to help you set up interconnects.

---

### Specifying bandwidth usage for a Data Mover interconnect

When you define a bandwidth schedule, you do so for each side of a Data Mover interconnect and it applies to all replication sessions that are using that interconnect. The bandwidth schedule allocates the interconnect bandwidth used on specific days, or specific times, or both, instead of using all available bandwidth at all times. By default, an interconnect provides all available bandwidth at all times for the interconnect. Therefore, any periods not covered by any of the configured intervals will use all of the bandwidth.

Define each time period (bandwidth limit) in most-specific to least-specific order. Start times run from 00:00 to 23:00 and end times run from 01:00 to 24:00. You cannot specify an end time lower than a start time. For example, a first entry of `MoTuWeThFr07:00-18:00/2000` and a second entry of `/8000` means use a limit of 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday; otherwise, use 8000 Kb/s.

---

**Note:** The bandwidth schedule executes based on Data Mover time, not Control Station time.

---

To configure a bandwidth schedule for a specified interconnect, use the `nas_cel -interconnect -bandwidth` option. Use the `nas_cel -interconnect -modify -bandwidth` option

to change an existing bandwidth schedule or specify a new schedule for an existing interconnect.

---

**Note:** A bandwidth limit of 0 means no data is sent over the interconnect. [Example 4 on page 40](#) shows the syntax.

---

To reset an existing bandwidth schedule to use the default of all available bandwidth for all days, specify two single quotes or two double quotes. For example:

```
nas_cel -interconnect -modify s2_s3 -bandwidth ''
nas_cel -interconnect -modify s2_s3 -bandwidth ""
```

You can set overlapping bandwidth schedules to specify smaller time intervals first and bigger time intervals next. The limit set for the first applicable time period is taken into account, so the order in which the schedule is created is important. For example, the following schedule syntax means that Monday at 9:30 A.M., the limit is 5000 Kb/s but Monday at 8:00 A.M., the limit is 7000 Kb/s:

```
Mo09:00-12:00/5000,MoTu08:00-17:00/7000,MoTuWeThFr06:00-18:00/10000
```

---

### Example 1

The following bandwidth schedule modifies interconnect s2\_s3 to use a limit of 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday; otherwise use a bandwidth schedule of 8000 Kb/s:

```
nas_cel -interconnect -modify s2_s3 -bandwidth
MoTuWeThFr07:00-18:00/2000,/8000
```

---

### Example 2

The following bandwidth schedule modifies interconnect s2\_s3 to use a limit of 2000 Kb/s from 9 A.M. to 2 P.M. Monday and Tuesday; and a limit of 5000 Kb/s Wednesday through Friday from 6 A.M. to 2 P.M.; otherwise use a bandwidth schedule of 8000 Kb/s:

```
nas_cel -interconnect -modify s2_s3 -bandwidth
MoTu09:00-14:00/2000,WeThFr06:00-14:00/5000,/8000
```

---

### Example 3

The following bandwidth schedule modifies interconnect s2\_s3 to use a limit of 2000 Kb/s from 9 A.M. to 2 P.M. Monday and Tuesday; a limit of 5000 Kb/s Wednesday and Thursday from 6 A.M. to 2 P.M.; a limit of 7000 Kb/s all day Friday; otherwise use a bandwidth schedule of 8000 Kb/s:

```
nas_cel -interconnect -modify s2_s3 -bandwidth
MoTu09:00-14:00/2000,WeTh06:00-14:00/5000,Fr00:00-24:00/7000,/8000
```

---

#### Example 4

The following bandwidth schedule modifies interconnect s2\_s3 to use no bandwidth, which in effect stops all data transfer over the interconnect on Monday, Tuesday and Friday from 8 A.M. to 10 P.M.:

```
nas_cel -interconnect -modify s2_s3 -bandwidth MoTuFr08:00-22:00/0
```

---

## How ongoing replication sessions work

VNX Replicator uses internal checkpoints to ensure availability of the most recent point-in-time copy. These internal checkpoints are based on SnapSure technology and are visible to replication users, but are not available for user management. This means you cannot perform operations on the checkpoint such as refresh, delete, and restore.

The replication session creation process involves the following steps, which are illustrated in [Figure 11 on page 42](#):

1. Network clients read and write to the source object through a Data Mover at the source site without interruption throughout this process. Network clients can read the destination object through a Data Mover at the destination site. However, in-process CIFS access will be interrupted during data transfer because the destination file system is unmounted/mounted as part of the synchronization operation. NFS clients should not experience an interruption.

---

**Note:** To avoid interruption throughout this process, use a checkpoint of the destination file system for client access at the destination site. The destination checkpoint should be accessed through its own share rather than by using the .ckpt virtual directory on the file system because access to .ckpt will be interrupted by the destination file system unmount/mount process.

---

2. A VNX administrator creates a replication session that supplies the information needed to set up and perform replication of a given source object. For example, the interconnect and an update policy.

---

**Note:** For remote replication, the same administrative user account with the nasadmin role must exist on both the source and destination VNXsystems.

---

3. The source-side interconnect specified for the replication session establishes the path between the source and destination. The session cannot be established unless both sides of the interconnect, source-side and destination, are available.
4. Replication creates two checkpoints for the source object.
5. For file system and VDM replication, a read-only destination object that is the same size as the source is created automatically as long as the specified storage is available. If the replication session identifies an existing destination object, the destination object, which must be the same size as the source and read-only, is reused.
6. Replication creates two checkpoints for the destination object.



7. Replication verifies that there is no common base established between the source and destination objects. For example, for file system or VDM replication, the destination file system or VDM does not exist.
8. A full (initial) copy is performed to transfer the data on the source to the destination object. If the user specifies a manual (on-demand) refresh, then data transfer will occur only when the user performs the refresh operation.
9. At the destination, the first checkpoint ckpt 1 is refreshed from the destination object to establish the common base with the source checkpoint.
10. After the common base is established, the second checkpoint ckpt 2 at the source is refreshed to establish the difference (delta) between the original source checkpoint and the latest point-in-time copy. Only the delta between the currently marked internal checkpoint and the previously replicated internal checkpoint is transferred to the destination by block ID. The transfer is called a differential copy.
11. After the data transfer is complete, replication uses the latest internal checkpoint taken on the destination to establish a new common base ckpt 2. The latest internal checkpoint contains the same data as the internal checkpoint on the source (ckpt 2) that was marked for replication.
12. Replication monitors the `max_time_out_of_sync` value (if set) to determine how frequently the destination is updated with source changes. Without an update policy, the user can perform an on-demand refresh.

---

**Note:** A replication session automatically starts when it is created.

---

[Chapter 6](#) and [Chapter 7](#) provide detailed procedures on how to create replication sessions.

Figure 11 on page 42 shows the basic elements of a single ongoing remote replication session, where the source and destination reside on different VNX systems.

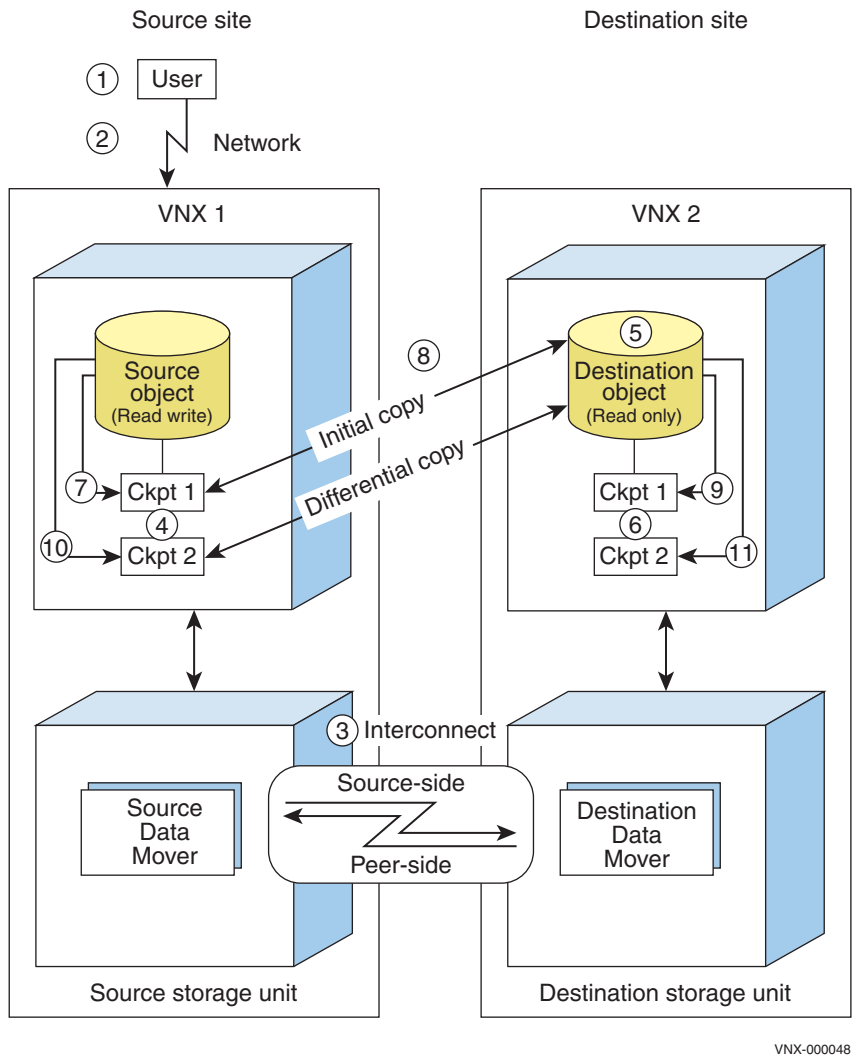


Figure 11. How ongoing replication sessions work

## How one-time file system copy sessions work

You create a one-time copy of a file system by using the `nas_copy` command. The source file system must be one of the following:

- ◆ Mounted read-only
- ◆ Checkpoint mounted read-only
- ◆ Writeable checkpoint mounted read-only

The basic steps in a copy session are as follows:

1. A user creates a one-time copy session by using the `nas_copy` command that supplies the information needed to set up and perform a copy of a given source file system.
2. The source-side interconnect specified for the copy session establishes the path between the source and destination. The session cannot be established unless both sides of the interconnect, source-side and destination (peer), are available.
3. For file system copy, replication creates an internal checkpoint for the source file system. No internal checkpoint is created for file system checkpoint or writeable checkpoint copy.
4. Replication creates the file system automatically on the read-only destination object that is the same size as the source as long as the specified storage is available. If the source is a read-only file system, it also creates an internal checkpoint for the file system.

For file system checkpoint copy, if the file system that the checkpoint is using already exists on the destination, it will create a new checkpoint with the naming convention `<source checkpoint name_replica#>`. If the file system does not exist yet, it will be created along with the checkpoint.

5. Copy begins:
  - When a file system or writeable checkpoint is copied, replication performs a full copy of the source to the destination file system.
  - When a file system checkpoint is copied, the copy is differential if an earlier checkpoint that serves as the common base exists (otherwise a full copy is performed) and the checkpoint appears on the destination.

---

**Note:** A differential copy is the difference (delta) between the original source checkpoint and the latest point-in-time copy. Only the delta between the file system checkpoint and the source file system is transferred to the destination. The transfer is called a differential copy.

---

6. After the data transfer is complete, replication deletes the session and the internal checkpoints created on the source and destination. For a checkpoint copy, the internal checkpoint created on the destination becomes a user checkpoint, which means it can be managed by the user.

---

**Note:** You can update (refresh) an existing destination checkpoint that has the same name as the copied checkpoint. You cannot refresh a loopback or local copy because the destination cannot create a checkpoint by the same name.

---

[Chapter 8](#) describes how to configure a one-time copy session.

---

## Updating the destination site with source changes

The update policy set for a replication session determines how frequently the destination is updated with source changes. You can define a `max_time_out_of_sync` value for a replication session or you can perform on-demand, manual updates.

The refresh operation handles on-demand updates of the destination side of a replication session based on changes to the source side. Alternatively, you can define an update policy for a replication session that automatically updates the destination based on the `max_time_out_of_sync` specified number of minutes in which the source and destination can be out of synchronization.

---

### Specifying a `max_time_out_of_sync` value

The `max_time_out_of_sync` value represents the elapsed time window within which the system attempts to keep the data on the destination synchronized with the data on the source. For example, if the `max_time_out_of_sync` is set to 10 minutes, then the content of, and latest checkpoint on, the destination should not be older than 10 minutes.

The destination could be updated sooner than this value. The source write rate, network link speed, and the interconnect throttle, when set, determine when and how often data is sent. Large bursts of writes on any file system that is sharing an interconnect or physical connection, or network problems between the source and destination can cause the `max_time_out_of_sync` setting to be exceeded.

---

### Manually updating the destination

The manual refresh option allows you to perform on-demand updates of the destination with the data on the source. When you change a replication session from a `max_time_out_of_sync` policy to a manual refresh, the system suspends the session. To resume the session, reset the `max_time_out_of_sync` value.

[Refresh the destination on page 132](#) describes in detail how to perform the refresh task.

---

## Stopping a replication session

You can temporarily stop an active replication session and leave replication in a condition that allows it to be started again. Stop allows you to temporarily stop a replication session, perform some action (mount a file system on a different Data Mover), and then start the replication session again by using a differential copy rather than a full data copy.

You may want to stop a replication session to:

- ◆ Mount the replication source or destination file system on a different Data Mover.
- ◆ Change the IP addresses or interfaces the interconnect is using for this replication session.

---

### How it works

The stop operation works as follows, depending on the type of session (local, remote, or loopback) and the stop `-mode` option selected.

To stop a local or remote replication session from the source VNX, select:

- ◆ Both — Stops both the source and destination sides of the session, when the destination is available.
- ◆ Source — Stops only the replication session on the source and ignores the other side of the replication relationship.

For a loopback replication session, this automatically stops both sides of the session.

To stop a local or remote replication session from the destination VNX, select:

- ◆ Destination — Stops the replication session on the destination side only.

For a loopback replication session, this automatically stops both sides of the session.

When you stop a replication session by using the Both or Destination mode options, and data transfer is in progress, the system will perform an internal checkpoint restore operation of the latest checkpoint on the destination side. This will undo any changes written to the file system as part of the current data transfer and bring the file system back to a consistent state. If the checkpoint restore fails, for example there was not enough SavVol space, the destination file system will remain in an inconsistent state and client-based access should be avoided.

[Stop a replication session on page 134](#) describes in detail how to perform this task.

---

## Starting a replication session

After you stop a replication session, only the start option can restart it. There are two primary use cases when starting a replication session:

- ◆ Start a session that was temporarily stopped and update the destination by using either a differential copy or a full data copy. For example, you stop the session to perform maintenance on the destination side and when done restart the session. This session is in the stopped state.
- ◆ Start a session that was automatically stopped due to a failover or switchover by using either a differential copy or a full data copy. This session was left in a failed over or switched over state.

When you start a replication session, you can change the interconnect and interfaces (source and destination) used in the session. You can specify a source and destination interface or allow the system to select it for you. If you allow the system to select it, the system will automatically verify that the source and destination interfaces can communicate. Otherwise, there is no check to verify communication.

You use the reverse option to start a replication session again after a failover or switchover. This will reverse the direction of the replication session, but not change the destination or source mount status (read-only or read/write).

When you start a replication in reverse, replication will only reverse one session in a one-to-many configuration.

---

## How it works

The start operation works as follows when the replication session is in a stopped state and not in a failed over or switched over state. [Starting a session after failover on page 51](#) describes how to start a replication session that is in a failed over state and [Starting a session after switchover on page 48](#) describes how to start a session that is in a switched over state:

1. A user starts the stopped replication session from the source side.
2. The local source-side Data Mover interconnect establishes the path to use between the source and destination. Any source or destination (peer) interface defined for the interconnect will be used unless the user specifies a specific source or destination interface to use for the replication session.
3. Replication verifies if there is an internal, common base checkpoint established between the source and destination objects. If there is a common base, replication will perform a restore on the destination from the common base and then start a differential copy. This will transfer all of the changes made to the source object since the replication session was stopped.
4. The update policy set for the replication session determines how frequently the destination is updated with the source changes. When you start a replication session, you can make changes to the policy. Otherwise, default values are used. For example, the `max_time_out_of_sync` default time is 10 minutes for file system replications, and 5 minutes for VDM replications.

---

## Start options

If you want to start a session and the destination object changed, you must specify either the `overwrite_destination` or `full_copy` option:

- ◆ `overwrite_destination` — Use this option when there is a common base established and the destination object changed. For example, you mounted the destination file system read/write, made changes, and you want to discard those changes.
- ◆ `full_copy` — Use this option to perform a full data copy of the source object to the destination object when there is no common base established. If there is no common base and you do not specify the `full_copy` option, the command will fail. If a common base exists, you do not have to specify the `full_copy` option unless you want the system to ignore the common base and perform a full copy.

If a common base exists and it contains different content than the destination, either:

- Specify both the `overwrite_destination` and `full_copy` options to discard the destination content completely and perform a full copy of the source object to the destination object.
- or
- Specify the `overwrite_destination` option without the `full_copy` option to restore the destination to the common base content and start a differential copy.

[Start a replication session on page 136](#) describes in detail how to perform this task.

---

## Reversing a replication session

You perform a reverse operation from the source side of a replication without data loss. This operation will reverse the direction of the replication session thereby making the destination read/write and the source read-only.

---

### How it works

This operation does the following:

1. Synchronizes the destination object with the source.
2. Mounts the source object read-only.
3. Stops replication.
4. Mounts the destination read/write.
5. Starts replication in reverse direction from a differential copy and with the same configuration parameters.

---

### Reverse in a one-to-many configuration

If you are running replication in a one-to-many configuration and you reverse one of the sessions, the source side goes into a cascading mode where the destination side from one of the multiple sessions becomes the source and the original source side becomes the destination and that source cascades out to other destination sessions.

---

### Reversing a VDM session

If you used VDMs for replication in your CIFS environment, remember to reverse the replication relationship of the VDM and user file systems in that VDM. Also, successful access to CIFS servers after a reverse requires that you maintain DNS, Active Directory, user mapping, and network support of the data recovery destination site. The VNX system depends on the DNS, Active Directory, user mapping, and network systems at your destination site to function correctly upon a reverse operation.

[Reverse the direction of a replication session on page 140](#) describes how to perform this task.

---

## Switching over a replication session

For test or migration purposes when the source is functioning and available, you can switch over the specified replication session to perform synchronization of the source and destination without data loss. You perform this operation on the source VNX only.

---

### How it works

The switchover operation does the following:

1. Synchronizes the destination object with the source.
2. Mounts the original source object as read-only.
3. Synchronizes the destination object with the source again.
4. Stops the replication.
5. Mounts the original destination object as read/write so that it can act as the new source object.

The replication direction remains from the original source to the destination.

6. When the original source site becomes available, start the session from the current destination side with the `-reverse` option. This reverses the direction of the session, but does not change the destination or source mount status (read-only or read/write).

The `-reverse` option is different from the replication `Reverse` operation, which will reverse the direction of a replication session and make the destination read/write and the source read-only. When you start a replication in reverse, replication will only reverse one session in a one-to-many configuration.

---

### Switching over a VDM session

Successful access to CIFS servers, when failed over, depends on the customer taking adequate actions to maintain DNS, Active Directory, user mapping, and network support of the data recovery (destination) site. VNX system depends on the DNS, Active Directory, user mapping, and network systems at your destination site to function correctly upon failover.

---

### Starting a session after switchover

When a replication session is switched over, the original source object becomes read-only, the original destination object becomes read/write and the session is stopped. The replication direction remains from the original source to the destination. When the original source site becomes available, start the session from the current destination side with the `-reverse` option to change the direction of the session and overwrite the original source with the data from the current source. Then perform a reverse operation to return the session to its original direction.

You can start a switched over replication from the source side, but before doing so you must change the mount status of the source object to read/write and the destination object to read-only.



---

### Example

This example explains how to start a replication after a switchover occurs between source file system fs1 and destination file system fs2:

1. After switchover, the fs1 file system becomes read-only and fs2 becomes read/write. The replication session is stopped and the replication direction remains as fs1 --> fs2.
2. The original fs2 destination is used as the source object until the original fs1 source is available.
3. When the original source becomes available, start the session from the destination side as follows:
  - a. Use the `-reverse` option. This will change the session direction to fs1 <-- fs2 [the original destination (the current source fs2) to the original source (the current destination fs1)].
  - b. (Optional) Use the `overwrite_destination` option to overwrite the original fs1 source object with the current source fs2 object data if, after switchover occurred, you changed the mount status of the original source object from read-only (which is the state replication puts it in during switchover) to read-write and made modifications to the original source.
4. Perform a replication Reverse operation to change the session back to the original direction fs1 --> fs2.

[Start a failed over or switched over replication on page 138](#) describes how to perform this task.

---

### Switchover in a one-to-many configuration

To start switched-over sessions when the source object is used in multiple replication sessions:

1. Select the replication session from which you want to resynchronize.
2. Start the replication session identified in step 1 from the original destination side and specify the `-reverse` option.
3. Reverse that replication session by using `nas_replicate -reverse` command.
4. If you switched over multiple sessions at one time, change the mount status of the destination objects in the remaining sessions. [Change the mount status of source or destination object on page 177](#) describes how to perform this task for each type of source object.
5. Start the remaining sessions from the original source side. When starting each session, ensure that you specify the `-overwrite_destination` option. Do not specify the `-reverse` option.

[Switch over a replication session on page 141](#) describes how to perform this task.

---

## Failing over a replication session

In a failover scenario, the source site has experienced a disaster and is unavailable. In response to this potential disaster scenario, you can perform a failover of the specified replication session, and the failover occurs with possible data loss. You perform a failover only from the destination VNX.

---

### How it works

The failover process works as follows:

1. Stops any data transfer that is in process.
2. Mounts the destination object as read/write so that it can serve as the new source object.
3. When the original source Data Mover becomes reachable, the source file system object is changed to read-only, and a VDM is changed to a mounted state.

The execution of the failover operation is asynchronous and will result in data loss if all the data is not transferred to the destination site prior to issuing the failover. After the source site is restored to service, start the replication session with the `-reverse` option to reverse the direction of the replication session but will not change the destination or source mount status (read-only or read/write).

The `-reverse` option is different from the replication Reverse operation, which will reverse the direction of a replication session and make the destination read/write and the source read-only.

---

**Note:** When you start a replication in reverse, replication will only reverse one session in a one-to-many configuration.

---

---

### Failing over VDM sessions

Successful access to CIFS servers, when failed over, depends on the customer taking adequate actions to maintain DNS, Active Directory, user mapping, and network support of the data recovery (destination) site. The VNX system depends on the DNS, Active Directory, user mapping, and network systems at your destination site to function correctly upon failover.

If the replication relationship involves a VDM, perform the failover in this order:

1. Fail over the VDM.

The system changes the VDM state from mounted to loaded on the destination site. If the source site is still available, the system attempts to change the VDM on the source site from loaded to mounted.

2. Fail over the user file system contained in that VDM.



The VNX system provides you with the ability to fail over a CIFS server and its associated file systems to a remote location. In a disaster, since VNX Replicator is an asynchronous solution, there might be some data loss. However, the file systems will be consistent (fsck is not needed). When using databases, additional application-specific considerations might be necessary to bring the database to a consistent state (for example, quiescing the database).

[Fail over a replication session on page 144](#) describes how to perform this task.

---

### Starting a session after failover

When a replication session fails over, the original source object becomes read-only, the original destination object becomes read/write and the session is stopped. The replication direction remains from the original source to the destination. When the original source site becomes available, start the session from the current destination side with the `-reverse` option to change the direction of the session and overwrite the original source with the data from the current source. Then perform a reverse operation to return the session to its original direction.

You can start a failed over replication from the source side, but before doing so you must change the mount status of the source object to read/write and the destination object to read-only.

---

### Example

This example explains how to start a replication after a failover occurs between source file system fs1 and destination file system fs2:

1. After failover, the fs1 file system becomes read-only and fs2 becomes read/write. The replication session is stopped and the replication direction remains as fs1 --> fs2.
2. The original fs2 destination is used as the source object until the original fs1 source is available.
3. When the original source becomes available, start the session from the destination side as follows:
  - a. Use the `-reverse` option. This will change the session direction to fs1 <-- fs2 [the original destination (the current source fs2) to the original source (the current destination fs1)].
  - b. Use the `overwrite_destination` option to overwrite the original fs1 source object with the current source object data.
4. Perform a replication Reverse operation to change the session back to the original direction fs1 --> fs2.

[Start a failed over or switched over replication on page 138](#) describes how to perform this task.

### Failover in a one-to-many configuration

EMC supports only one failed-over replication session per source object. If you fail over more than one session for a given source object, you can save only the changes from one of the failed-over replication sessions.

Before you start, ensure that the source site is now available.

To recover from this:

1. Select the failed over session from which you want to save the changes and resynchronize
2. Start that replication session and specify the `-overwrite_destination` and `-reverse` option.
3. Reverse the replication session by using `nas_replicate -reverse`.
4. If you failed over multiple sessions, change the mount status of the destination objects to read-only in the remaining sessions.
5. Start the remaining sessions from the original source side. When starting each session, ensure that you specify `-overwrite_destination`. Do not specify the `-reverse` option.

### When to use failover, switchover, and reverse

[Table 4 on page 52](#) summarizes when you should use the failover, switchover, and reverse operations.

**Table 4. Failover, switchover, and reverse operations**

Operation	Use if	What happens	Site that must be available
Failover	The source site is totally corrupt or unavailable.	In response to a potential disaster scenario, performs a failover of the specified replication session with data loss. Run this command from the destination VNX only.  This command cancels any data transfer that is in process and marks the destination object as read/write so that it can serve as the new source object. When the original source Data Mover becomes reachable, the source object is changed to read-only.	Destination

Table 4. Failover, switchover, and reverse operations (*continued*)

Operation	Use if	What happens	Site that must be available
Switchover	The source site is still available.	<p>For test or migration purposes, switches over the specified replication relationship and performs synchronization of the source and destination without data loss. Use this option if the source site is available but you want to activate the destination file system as read/write.</p> <p>Execute this command from the source VNX only. This command stops replication, mounts the source object as read-only, and mounts the destination object as read/write so that it can act as the new source object.</p> <p>Unlike a reverse operation, a switchover operation does not start replication.</p>	Source and destination
Reverse	The source site is still available.	<p>Execute this command from the source VNX only.</p> <p>This command synchronizes the destination object with the source, mounts the source object read-only, stops the replication, and mounts the destination read/write. Then, unlike switchover, starts replication in reverse direction from a differential copy by using the same configuration parameters originally established for the session.</p>	Source and destination

## Deleting a replication session

You would delete a replication session when you no longer want to replicate the file system or VDM.

**Note:** Deleting replication does not delete the underlying source or destination objects. However, the internal checkpoints are deleted as part of the operation.

The delete operation works as follows:

When deleting a local or remote replication session from the source VNX, select:

- ◆ Both — Deletes both source and destination sides of the session, as long as the destination is available.
- ◆ Source — Deletes only the replication session on the source and ignores the other side of the replication relationship.

For a loopback replication session, this automatically deletes both sides of the session.

When deleting a remote or local replication session from the destination VNX, select:

- ◆ Destination — Deletes the replication session on the destination side only. Use this option to delete a session from the destination side that could not be deleted using the -both option because the destination side was unavailable.

For a loopback replication session, this automatically deletes both sides of the session.

When you delete a replication session by using the Both or Destination mode options, and data transfer is in progress, the system will perform an internal checkpoint restore operation of the latest checkpoint on the destination side. This will undo any changes written to the file system as part of the current data transfer and bring the file system back to a consistent state. If the checkpoint restore fails, for example there was not enough SavVol space, the destination file system will remain in an inconsistent state and client-based access should be avoided.

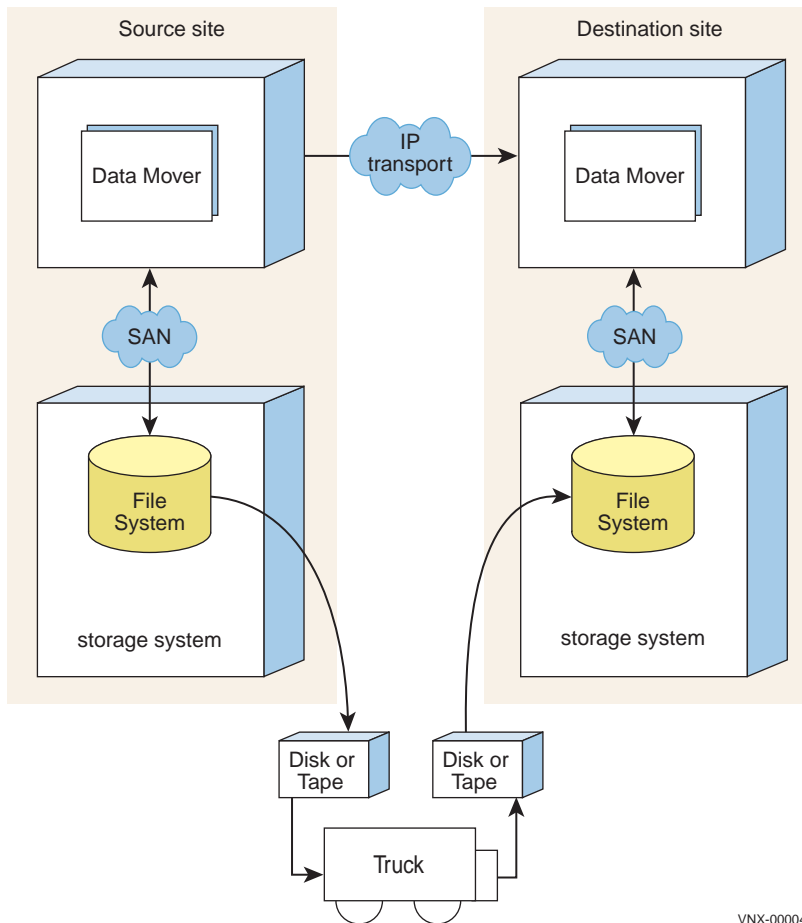
[Delete a replication session on page 133](#) describes how to perform this task.

---

## Performing an initial copy by using disk or tape

Copying the baseline source file system from the source to the destination site over the IP network can be a time-consuming process. You can use an alternate method by copying the initial checkpoint of the source file system, backing it up to a disk array or tape drive,

and transporting it to the destination site, as shown in [Figure 12 on page 55](#). This method is also known as silvering.



VNX-000047

**Figure 12. Physical transport of data**

---

**Note:** EMC recommends that the initial copy of the root file system for a VDM to be performed over the IP network.

---

### Disk transport method

If the source file system holds a large amount of data, the initial copy of the source to the destination file system can be time consuming to move over the IP network. Preferably, move the initial file system copy by disk, instead of over the network.

When using the disk transport method, note the following:

- ◆ Three VNX systems are required:

- VNX A located at the source site
  - VNX B, a small swing VNX used to transport the baseline copy of the source file system to the destination site
  - VNX C located at the destination site
- ◆ You must set up communication between each VNX system (A to B and B to C) before creating replication sessions. [Chapter 4](#) describes how to configure communication between VNX systems.
  - ◆ You must set up communication between Data Movers on each VNX system (A to B, B to C, and A to C) before creating the copy sessions. [Chapter 5](#) explains how to create communication between Data Movers.
  - ◆ You should ensure that there is enough disk space on all of the VNX systems.

[Perform an initial copy by using the disk transport method on page 178](#) describes how to perform this task.

---

### Tape transport method

If the source file system contains a large amount of data, the initial copy of the source file system to the destination file system can be time consuming to move over the IP network. Moving the initial copy of the file system by backing it up to tape, instead of over the network, is preferable.

---

**Note:** This special backup is used only for transporting replication data.

---

This method is not supported for VDM replications.



Backing up file systems from a Unicode-enabled Data Mover and restoring to an ASCII-enabled Data Mover is not supported. I18N mode (Unicode or ASCII) must be the same on both the source and destination Data Movers.

### Before you begin

When using this transport method, note the following:

- ◆ You must have a valid NDMP-compliant infrastructure on both source and destination VNX systems.
- ◆ At most, four NDMP backups can be active at any given time.
- ◆ If failover occurs during backup or restore, the Data Management Application (DMA), such as EMC NetWorker®, requires that the tape backup and restore process be restarted.
- ◆ The replication signature is backed up for a tape backup, and after restore on the destination side, this signature is automatically applied.

### How tape transport works



The tape transport process works as follows:

1. On the source site, create a file system replication session with the `-tape_copy` option. This option creates the replication session, and then stops it to enable an initial copy using a physical tape.
2. From the NDMP DMA host, create a tape backup of the source file system.
3. The DMA sets up the tape and sends the information to start backup to the Data Mover. The Data Mover begins the backup and starts copying data from disk to tape.
4. Move the tape and all relevant DMA catalogs to the destination.

---

**Note:** Be aware that changes accumulate in the checkpoint while the session is stopped. Ensure that the high water mark and auto extend options are set appropriately.

---

5. Restore the backup image from the tape to the destination site:
  - The DMA sets up the tape and sends the information to begin the restore to the Data Mover. The Data Mover begins the restore. If any kind of tape management is needed, such as a tape change, the Data Mover will signal the DMA.
  - If an unrecoverable volume-write or tape-read error occurs or the destination file system specified is not valid, the restore fails and error messages are logged to the server log and the DMA log file.
6. Start the replication session created in step 1. The system detects the common base checkpoint created in step 1 and starts the differential data transfer.

[Perform an initial copy by using the tape transport method on page 183](#) describes how to perform this task.

---

## Planning considerations

These topics help you plan and implement VNX Replicator . The topics include:

- ♦ [Controlling replication sessions with replication policies on page 58](#)
- ♦ [Interconnect setup considerations on page 58](#)
- ♦ [Accommodating network bandwidth on page 59](#)
- ♦ [Determining the number of replications per Data Mover on page 59](#)
- ♦ [Using dual Control Stations on page 60](#)
- ♦ [Infrastructure planning considerations on page 61](#)
- ♦ [Configuring events for VNX Replicator on page 62](#)

---

### Controlling replication sessions with replication policies

You can set policies to control replication sessions. Most of these policies can be established for one replication session by using the `nas_replicate` command or all replication sessions on a Data Mover by using `nas_cel` or a parameter. VNX Replicator has policies to:

- ◆ Control how often the destination object is updated by using the `max_time_out_of_sync` setting.
- ◆ Control throttle bandwidth for a Data Mover interconnect by specifying bandwidth limits on specific days, specific hours, or both.

---

### Interconnect setup considerations

Before you set up interconnects, consider the following planning information:

- ◆ To prevent potential errors during interface selection, especially after a failover/switchover, specify the same source and destination interface lists when configuring each side of the interconnect.
- ◆ When multiple subnets exist for a Data Mover, ensure that the network routing table contains a routing entry to the source Data Mover's routing table to the specified destination and gateway address for the subnet to be used. Otherwise, all network traffic will go to the default router.
- ◆ When defining a Data Mover interconnect, all of the source interfaces must be able to communicate with all of the destination interfaces in both directions. [Verify interface network connectivity on page 70](#) provides more details on how to verify network connectivity for interconnects.
- ◆ If you plan to use IP addresses, make sure the corresponding destination interface list uses the same IPv4/IPv6 network protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol (IPv4/IPv6). For each network protocol type specified in the source interface list, at least one interface from the same type must be specified in the destination list and vice versa. For example, if the source interface list includes one or more IPv6 addresses, the destination interface list must also include at least one IPv6 address. Validation fails for any interconnect that does not have interfaces from the same protocol in both source and destination lists.
- ◆ If you plan to use name service interface names when defining an interconnect, the interface name must already exist and the name must resolve to a single IP address (for example, by using a DNS server). In addition, name service interface names should, but are not required to be fully qualified names.

---

**Note:** Name service interface names are not the same as VNX network device names (for example, `cge0`). They are names that are resolved through a naming service (local host file, DNS, NIS, or LDAP).

---

- ◆ When defining a bandwidth schedule for a Data Mover interconnect, note that the schedule executes based on Data Mover time, and not Control Station time.

---

### Accommodating network bandwidth

Consider the following Quality of Service (QoS) and network suggestions suitable to your bandwidth requirements:

- ◆ Use dedicated network devices for replication to avoid an impact on users.
- ◆ Apply QoS policy by creating a bandwidth schedule that matches the bandwidth of one network to another. For instance, even if you have a 1 MB/s line from A to B and want to fill the pipe, define a bandwidth schedule at 1 MB/s so the line will not flood, causing packets to drop. Bandwidth can be adjusted based on the number of concurrent transfers and available bandwidth.

Configure the bandwidth schedule to throttle bandwidth for a replication session (or bandwidths for different sessions) on a Data Mover using the `nas_cel -interconnect` command. The *EMC VNX Command Line Interface for File* further details the `nas_cel -interconnect` command.

- ◆ If you do not expect significant packet loss and are in a high-window, high-latency environment, use the `fastRTO` parameter. `fastRTO` determines which TCP timer to use when calculating retransmission time out. The TCP slow timer (500 ms) is the default, causing the first time-out retransmission to occur in 1–1.5 seconds. Setting `fastRTO` to 1 sets the TCP fast timer (200 ms) for use when calculating retransmission time out, causing the first time out to occur in 400–600 ms. Changing the retransmission time also affects the interval a connection stays open when no response is heard from the remote link.

Use `server_param <movername> -facility tcp fastRTO=1` to configure the setting.

---

**Note:** This setting may actually increase network traffic. Therefore, make the change cautiously, recognizing it might not improve performance.

---

The *Parameters Guide for VNX for File* provides more information.

---

### Determining the number of replications per Data Mover

Determine the maximum number of replication sessions per Data Mover based on your configuration, such as the WAN network bandwidth and the production I/O workload. This number is also affected by whether you are running both SnapSure and VNX Replicator on the same Data Mover. Both of these applications share the available memory on a Data Mover.

Each internal checkpoint uses a file system ID, which will effect the file system limit even though these internal checkpoints do not count toward the user checkpoint limit.

For all configurations, there is an upper limit to the number of replications allowed per Data Mover. [Restrictions on page 13](#) details the current number of replications allowed per Data Mover.

If you plan to run loopback replications, keep in mind that each loopback replication counts as two replication sessions since each session encapsulates both outgoing and incoming replications.

You should verify that the Data Mover can handle all replication sessions and production I/Os. You can also monitor memory usage and CPU usage by using the `server_sysstat` command. This command shows total memory utilization, not just VNX Replicator and SnapSure memory usage.

---

**Note:** Use Unisphere to monitor memory and CPU usage by creating a new notification on the Monitoring > Notifications > Data Mover Load tab.

---

## Using dual Control Stations

If your VNX is configured with dual Control Stations you must use the IP aliasing feature. If you do not, replications will not work if the primary Control Station fails over to the secondary Control Station.

### IP aliasing

Configuring IP aliasing for the Control Station enables communication with the primary Control Station using a single IP address regardless of whether the primary Control Station is running in slot 0 or slot 1.

The following guidelines apply to this feature:

- ◆ When using replication for the first time, or on new systems, configure IP alias first, then use IP alias in the `nas_cel -create -ip <ipaddr>` command.
- ◆ For existing systems with existing replication sessions, the current slot\_0 primary Control Station IP address must be used. For example:

```
# nas_config -IPalias -create 0
Do you want slot_0 IP address <10.6.52.154>
as your alias? [yes or no]: yes
Please enter a new IP address for slot_0: 10.6.52.95
Done
#
```

- ◆ If the Control Station fails over while a replication task is running, use `nas_task` to check the status of the task. If the task failed, re-issue the replication command.
- ◆ When IP alias is deleted using the `nas_config -IPalias -delete` command, the IP address of the primary or the secondary Control Station is not changed. Changes to the IP address of the primary or the secondary Control Station must be done separately. Replication depends on communication between the source Control Station and the destination Control Station. When IP alias is used for replication, deleting the IP alias breaks the communication. The IP address which was used as

the IP alias must be restored on the primary Control Station to restore the communication.

- ◆ While performing a NAS code upgrade, do not use an IP alias to log in to the Control Station.

*VNX System Operations* provides details for setting up IP aliasing.

*EMC VNX Command Line Interface for File* provides details on the `nas_config` command.

---

### Infrastructure planning considerations

Note the following configuration considerations:

- ◆ Carefully evaluate the infrastructure of the destination site by reviewing items such as:
  - Subnet addresses
  - Unicode configuration
  - Availability of name resolution services; for example, WINS, DNS, and NIS
  - Availability of WINS/PDC/BDC/DC in the right Microsoft Windows NT, or Windows Server domain
  - Share names
  - Availability of user mapping, such as the EMC Usermapper for VNX systems
- ◆ The CIFS environment requires more preparation to set up a remote configuration because of the higher demands on its infrastructure than the Network File System (NFS) environment. For example, authentication is handled by the domain controller. For the CIFS environment, you must perform mappings between the usernames/groups and UIDs/GIDs with the Usermapper or local group/password files on the Data Movers. [Appendix A](#) provides more details on setting up the CIFS replication environment.
- ◆ The destination file system can only be mounted on one Data Mover, even though it is read-only.
- ◆ At the application level, as well as the operating system level, some applications may have limitations on the read-only destination file system due to caching and locking.
- ◆ If you are planning on enabling international Unicode character sets on your source and destination sites, you must first set up translation files on both sites before starting Unicode conversion on the source site. *Using International Character Sets on VNX for File* covers this consideration.
- ◆ VNX's FileMover feature supports replicated file systems. This is described in *Using VNX FileMover*.
- ◆ VNX file-level retention capability supports replicated file systems. *Using VNX File-Level Retention* provides additional configuration information.

---

### Configuring events for VNX Replicator

You can configure your VNX system to perform an action when specific replication events occur. These are called notifications that originate from the system and can include, but are not limited to the following:

- ◆ Logging the event in a log file
- ◆ Sending an email
- ◆ Generating a Simple Network Management Protocol (SNMP) trap

For example, if the `max_time_out_of_sync` value is set to 10 minutes and the data transfer does not complete within this value, you can configure an email notification alerting you that the SLA was not met.

System event traps and email notifications are configured by the user. *Configuring Events and Notifications on VNX for File* describes how to manage these notifications.

## Upgrading from previous versions

This section describes the upgrade process when upgrading from a previous version of Celerra Network Server to VNX for file version 7.1 when using VNX Replicator.

The tasks to upgrade from previous versions are:

- ◆ [Upgrade from a previous release on page 64](#)
- ◆ [Enable VNX Replicator on page 64](#)

## Upgrade from a previous release

When performing an upgrade from Celerra Network Server 6.0 to VNX for file version 7.1, no action needs to be taken related to any existing VNX Replicator replication sessions.

### Upgrading to manage remote replications with a global user account

You must upgrade both the source and destination systems to version 7.1 to configure and manage remote replication sessions with a storage domain global user account. To synchronize local accounts between the source and destination, set up a global account across both systems to successfully replicate from one version 7.1 system to another.

## Enable VNX Replicator

After you upgrade, you must enable the VNX Replicator license.

Action
To install the license for the VNX Replicator software feature, type:
<pre>\$ nas_license -create replicatorv2</pre>



## Configuring communication between VNX systems

The tasks to set up communication between VNX systems are:

- ◆ [Prerequisites on page 66](#)
- ◆ [Set up communication on the source on page 66](#)
- ◆ [Set up communication on the destination on page 67](#)
- ◆ [Verify communication between VNX systems on page 68](#)

## Prerequisites

Before creating a replication session for remote replication, you must establish the trusted relationship between the source and destination VNX systems in your configuration.

**Note:** The communication between VNX Control Stations uses HTTPS.

The procedures in this section require the following:

- ◆ The systems are up and running and IP network connectivity exists between the Control Stations of both VNX systems. Verify whether a relationship exists by using the `nas_cel -list` command.
- ◆ The source and destination Control Station system times must be within 10 minutes of each other. Take into account time zones and daylight savings time, if applicable.
- ◆ The same 6–15 characters passphrase must be used for both VNX systems.

Replication sessions are supported between VNX systems that use version 7.0 and later and legacy Celerra systems that use version 6.0 and later. For legacy Celerra systems that use version 5.6, only versions later than 5.6.47.0 are supported for replication with legacy Celerra systems that use version 6.0 and later, and VNX systems that use version 7.0 and later.

### Sample information

[Table 5 on page 66](#) shows information about the source and destination sites used in the examples in this section.

**Table 5. Sample information**

Site	VNX system name	IP address
Source site	cs100	192.168.168.114
Destination site	cs110	192.168.168.102

## Set up communication on the source

This section describes the process of establishing communication between source and destination sites.

### Action

On a source VNX system, to establish the connection to the destination VNX system in the replication configuration, use this command syntax:

```
$ nas_cel -create <cel_name> -ip <ip> -passphrase <passphrase>
```

where:

Action
<p><code>&lt;cel_name&gt;</code> = name of the remote destination VNX system in the configuration</p> <p><code>&lt;ip&gt;</code> = IP address of the remote Control Station in slot 0</p> <p><code>&lt;passphrase&gt;</code> = the secure passphrase used for the connection, which must have 6–15 characters and be the same on both sides of the connection</p> <p>Example:</p> <p>To add an entry for the Control Station of the destination VNX system, cs100, from the source VNX system cs100, type:</p> <pre>\$ nas_cel -create cs110 -ip 192.168.168.10 -passphrase nasadmin</pre>
Output
<pre>operation in progress (not interruptible)... id           = 1 name        = cs110 owner       = 0 device      = channel     = net_path    = 192.168.168.10 celerra_id  = APM000420008170000 passphrase  = nasadmin</pre>

## Set up communication on the destination

This procedure describes how to set up communication on the destination VNX system to the source VNX system.

Action
<p>On a destination VNX system, to set up the connection to the source VNX system, use this command syntax:</p> <pre>\$ nas_cel -create &lt;cel_name&gt; -ip &lt;ip&gt; -passphrase &lt;passphrase&gt;</pre> <p>where:</p> <p><code>&lt;cel_name&gt;</code> = name of the remote source VNX system in the configuration</p> <p><code>&lt;ip&gt;</code> = the IP address of the remote Control Station in slot 0</p> <p><code>&lt;passphrase&gt;</code> = the secure passphrase used for the connection, which must have 6–15 characters and be the same on both sides of the connection</p> <p>Example:</p> <p>To add the Control Station entry of a source VNX system, cs100, from the destination VNX system cs110, type:</p> <pre>\$ nas_cel -create cs100 -ip 192.168.168.12 -passphrase nasadmin</pre>

**Output**

```
operation in progress (not interruptible)...
id           = 2
name        = cs100
owner       = 0
device      =
channel     =
net_path    = 192.168.168.12
celerra_id  = APM000417005490000
passphrase  = nasadmin
```

## Verify communication between VNX systems

Perform this procedure at both the source and destination sites to check whether the VNX system can communicate with one another. Perform this task for remote replication only.

**Action**

To verify whether the source and destination sites can communicate with each other, at the source site, type:

```
$ nas_cel -list
```

**Output**

id	name	owner	mount_dev	channel	net_path	CMU
0	cs100	0			192.168.168.114	APM000340000680000
1	eng168123	201			192.168.168.123	APM000437048940000
3	eng16853	501			192.168.168.53	APM000183501737000
5	cs110	503			192.168.168.102	APM000446038450000

**Note:** The sample output shows that the source site can communicate with the destination site, cs110.

**Action**

To verify whether source and destination sites can communicate with each other, at the destination site, type:

```
$ nas_cel -list
```

**Output**

id	name	owner	mount_dev	channel	net_path	CMU
0	cs110	0			192.168.168.10	APM000446038450000
2	cs100	501			192.168.168.12	APM000340000680000

**Note:** The sample output shows that the destination site can communicate with the source site, cs100.

## Configuring communication between Data Movers

The tasks to set up Data Mover-to-Data Mover communication are:

- ◆ [Prerequisites on page 70](#)
- ◆ [Verify interface network connectivity on page 70](#)
- ◆ [Set up one side of an interconnect on page 74](#)
- ◆ [Set up the peer side of the interconnect on page 76](#)
- ◆ [Validate interconnect information on page 77](#)

## Prerequisites

Before you set up Data Mover interconnects, review the following:

- ◆ [Data Mover interconnect on page 34](#)
- ◆ Interconnect setup considerations in [Planning considerations on page 57](#)

The tasks to initially set up Data Mover-to-Data Mover communication for VNX Replicator involve the creation of a Data Mover interconnect. You must create an interconnect between a given Data Mover on one VNX system and its peer Data Mover that may reside on the same VNX system or a remote VNX system. The interconnect is only considered established after you set it up from both sides.

These tasks are required for local and remote replication. A loopback interconnect is established and named automatically for each Data Mover on the local VNX system.

You configure only one interconnect between the same Data Mover pair.

The interconnect between Data Movers is HTTPS in anonymous mode. This HTTPS setting indicates a secure HTTP connection and cannot be configured by user.

After you configure an interconnect for a Data Mover, you cannot rename that Data Mover without deleting and reconfiguring the interconnect and reestablishing replication sessions. [Chapter 13](#) provides more information about how to rename a Data Mover that has existing interconnects.

## Verify interface network connectivity

All source interfaces included in the interconnect for a Data Mover must be able to communicate to all interfaces on the destination Data Mover in both directions.

For example, if an interconnect is defined between local Data Mover server\_2, which has two IP addresses listed for source interfaces (192.168.52.136 and 192.168.53.137), and peer Data Mover server\_3, which has two IP address listed for the destination interfaces (192.168.52.146 and 192.168.53.147), then all of the interfaces must be able to communicate in both directions as follows:

192.168.52.136 to 192.168.52.146

192.168.52.136 to 192.168.53.147

192.168.53.137 to 192.168.52.146

192.168.53.137 to 192.168.53.147

192.168.52.146 to 192.168.52.136

192.168.52.146 to 192.168.53.137

192.168.53.147 to 192.168.52.136

192.168.53.147 to 192.168.53.137

You use the `server_ping` command with the `-interface` option to verify connectivity between every interface you plan to include on the source and destination Data Movers. The `-interface` option requires that the interface name be used instead of the IP address.

If no communication exists, use the Troubleshooting section of the *Configuring and Managing Networking on VNX* document to ensure that the interfaces are not down or that there are no other networking problems. If there are no other networking problems, create a static host route entry to the Data Mover routing table.

Table 6 on page 71 contains the sample information used to demonstrate this procedure.

**Table 6. Sample information**

Site	System name	Data Mover	Interface IP	Interface name	Device
Source	cs100	server_2 (local)	192.168.52.136	cs100-20	cge0
			192.168.53.137	cs100-21	cge1
			192.168.168.138	cs100-22	cge2
Destination	cs110	server_3 (peer)	192.168.52.146	cs110-30	cge0
			192.168.53.147	cs110-31	cge1
			192.168.168.148	cs110-32	cge2

The naming convention used in this section for interface names is: < VNX system name>-<server#><device#>. For example, the interface name for device `cge0` on source Data Mover `server_2` with IP address `192.168.52.136` is `cs100-20`:

1. To verify that communication is established from the source and destination interfaces, use this command syntax:

```
# server_ping <movername> -interface <interface> <ip_addr>
```

where:

<movername> = source Data Mover name

<interface> = interface name on the source Data Mover

<ip\_addr> = IP address of the interface on the destination Data Mover for which you are verifying network connectivity

Example:

To verify that `192.168.52.146` is accessible through interface `cs100-20` on source Data Mover `server_2`, type:

```
# server_ping server_2 -interface cs100-20 192.168.52.146
```

Output:

```
server_2 : 192.168.52.146 is alive, time= 0ms
```

The output shows that there is network connectivity from source interface cs100-20 with IP address 192.168.52.136 to IP address 192.168.52.146 on the destination.

- Repeat step 1 to verify network connectivity between the remaining source and destination interfaces.

Example:

To verify network connectivity, between the remaining source and destination interfaces, type:

```
# server_ping server_2 -interface cs100-20 192.168.53.147
```

Output:

```
server_2 : 192.168.53.147 is alive, time= 3 ms
```

```
# server_ping server_2 -interface cs100-21 192.168.53.147
```

Output:

```
server_2 : 192.168.53.147 is alive, time= 0 ms
```

```
# server_ping server_2 -interface cs100-21 192.168.52.146
```

Output:

```
server_2 :
Error_6: server_2 : No such device or address
no answer from 192.168.52.146
```

The output shows that there is no network connectivity from source interface cs100-21 with IP address 192.168.53.137 to IP address 192.168.52.146 on the destination.

---

**Note:** Use the Troubleshooting section of the *Configuring and Managing Networking on VNX* document to ensure that the interfaces are not down or that there are no other networking problems. If there are no other networking problems, go to step 3 and create a static host route entry to 192.168.52.146 by using gateway 192.168.53.1.

---

- If no communication is established and you have verified that no other networking problems exist, add a routing entry to the source Data Mover's routing table to the specified destination and gateway address, using this command syntax:

```
# server_route <movername> -add host <dest> <gateway>
```

where:

<movername> = name of the Data Mover

<dest> = specific host for the routing entry

<gateway> = the network gateway to which packets should be addressed

Example:



To add a static host route entry to 192.168.52.146 by using the network gateway 192.168.53.1, type:

```
# server_route server_2 -add host 192.168.52.146 192.168.53.1
```

Output:

```
server_2 : done
```

To verify that the communication is now established, type:

```
# server_ping server_2 -interface cs100-21 192.168.52.146
```

Output:

```
server_2 : 192.168.52.146 is alive, time= 19ms
```

4. On the destination system, to verify that communication is established from the destination to the source interfaces, use this command syntax:

```
# server_ping <movername> -interface <interface> <ip_addr>
```

where:

<movername> = destination Data Mover name

<interface> = interface name on the destination Data Mover

<ip\_addr> = IP address of the interface on the source Data Mover for which you are verifying network connectivity

Example:

To verify that communication is established from destination interface cs110-30 with IP address 192.168.52.146 to IP address 192.168.52.136 on the source, type:

```
# server_ping server_3 -interface cs110-30 192.168.52.136
```

Output:

```
server_3 : 192.168.52.136 is alive, time= 0 ms
```

The output shows that there is network connectivity from destination interface cs110-30 with IP address 192.168.52.146 to IP address 192.168.52.136 on the source.

5. Repeat step 4 to verify network connectivity from the remaining destination and source interfaces.

Example:

To verify network connectivity from the remaining interfaces, type:

```
# server_ping server_3 -interface cs110-30 192.168.53.137
```

Output:

```
server_3 : 192.168.53.137 is alive, time= 0 ms
```

```
# server_ping server_3 -interface cs110-31 192.168.53.137
```

Output:

```
server_3 : 192.168.53.137 is alive, time= 0 ms
```

```
# server_ping server_3 -interface cs110-31 192.168.52.136
```

Output:

```
server_3 :  
Error 6: server_3 : No such device or address  
no answer from 192.168.52.136
```

The output shows that there is no network connectivity from destination interface cs110-31 with IP address 192.168.53.147 to IP address 192.168.52.136 on the source.

---

**Note:** Use the Troubleshooting section of the *Configuring and Managing Networking on VNX* document to ensure that the interfaces are not down or that there are no other networking problems. If there are no other networking problems, create a static host route entry to 192.168.52.136 by using gateway 192.168.53.1.

---

6. If no communication is established and you have verified that no other networking problems exist, repeat step 3 and add a routing entry to the destination Data Mover's routing table to the specified destination and gateway address by using this command syntax:

```
# server_route <movername> -add host <dest> <gateway>
```

where:

<movername> = name of the Data Mover

<dest> = specific host for the routing entry

<gateway> = the network gateway to which packets should be addressed

Example:

To add a static host entry to 192.168.52.136 using the network gateway 192.168.53.1, type:

```
# server_route server_3 -add host 192.168.52.136 192.168.53.1
```

Output:

```
server_3 : done
```

To verify that the host route entry resolved the communication problem, type:

```
# server_ping server_3 -interface cs110-31 192.168.52.136
```

Output:

```
server_3 : 192.168.52.136 is alive, time= 35 ms
```

You can now use all of the interfaces on the local system and the interconnect created on the remote system. After you set up both sides of the interconnects, validate the interconnects to further verify network connectivity.

---

## Set up one side of an interconnect

Administrator privileges are required to execute this command.

**Action**

To set up one side of the interconnect on a given VNX system, use this command syntax:

```
$ nas_cel -interconnect -create <name> -source_server <movername>
-destination_system {<cel_name>|id=<cel_id>} -destination_server <movername>
-source_interfaces {<name_service_interface_name>|ip=<ipaddr>}
[, {<name_service_interface_name>|ip=<ipaddr>}, ...]
-destination_interfaces {<name_service_interface_name>|ip=<ipaddr>}
[, {<name_service_interface_name>|ip=<ipaddr>}, ...] [-bandwidth
<bandwidthSched>]
```

where:

<name> = name of the interconnect

<movername> = name of the Data Mover, first for the source side of the interconnect and then for the destination (peer) side of the interconnect

<cel\_name> = name of the destination (peer) VNX system

<cel\_id> = ID of the destination (peer) VNX system

<name\_service\_interface\_name> = name of a network interface, source or destination, that can be resolved to a single IP address (for example, by a DNS server)

<ipaddr> = the IP address (IPv4 or IPv6) of an interface, in dot notation

**Note:** Ensure that the destination interface list uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

<bandwidthSched> = a schedule with one or more comma-separated entries to specify bandwidth limits on specific days, specific hours, or both, listed most specific to least specific by using the syntax [[Su|Mo|Tu|We|Th|Fr|Sa]][HH:00-HH:00][Kb/s],[ <next\_entry>],[...]

**Note:** If no bandwidth schedule is specified, all available bandwidth is used at all times.

Example:

To set up an interconnect between Data Mover 2 on system cs100 and destination Data Mover 3 on peer system cs110, and specify a bandwidth schedule that limits bandwidth to 8000 Kb/s on Monday, Tuesday, and Wednesday from 7 A.M. to 2 P.M., at the source site, type:

```
$ nas_cel -interconnect -create cs100_s2s3 -source_server server_2
-destination_system cs110 -destination_server server_3 -source_interfaces
ip=192.168.52.136,ip=192.168.53.137,ip=192.168.168.138 -destination_interfaces
ip=192.168.52.146,ip=192.168.53.147, ip=192.168.168.148 -bandwidth
MoTuWe07:00-14:00/8000
```

**Output**

```

operation in progress (not interruptible)...
id = 20005
name = cs100_s2s3
source_server = server_2
source_interfaces = 192.168.52.136, 192.168.53.137,
                    192.168.168.138
destination_system = cs110
destination_server = server_3
destination_interfaces = 192.168.52.146, 192.168.53.147,
                        192.168.168.148
bandwidth schedule = MoTuWe07:00-14:00/8000
crc enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.

```

## Set up the peer side of the interconnect

Each site must be configured for external communication and be active.

**Action**

To set up the peer side of the interconnect from the remote VNX system or from the same system with a different Data Mover for local replication, use this command syntax:

```

$ nas_cel -interconnect -create <name> -source_server <movername>
  -destination_system {<cel_name>|id=<cel_id>} -destination_server <movername>
  -source_interfaces {<name_service_interface_name>|ip=<ipaddr>}
  [, {<name_service_interface_name>|ip=<ipaddr>}, ...]
  -destination_interfaces {<name_service_interface_name>|ip=<ipaddr>}
  [, {<name_service_interface_name>|ip=<ipaddr>}, ...] [-bandwidth
  <bandwidthSched>]

```

where:

<name> = name of the interconnect

<movername> = name of the Data Mover, first for the local side of the interconnect and then for the destination (peer) side of the interconnect

<cel\_name> = name of the destination (peer) VNX system

<cel\_id> = ID of the destination (peer) VNX system

<name\_service\_interface\_name> = name of a network interface, source or destination, that can be resolved to a single IP address (for example, by a DNS server)

<ipaddr> = the IP address (IPv4 or IPv6) of an interface, in dot notation

**Note:** Ensure that the source interface list uses the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

Action
<p><code>&lt;bandwidthSched&gt;</code> = a schedule with one or more comma-separated entries to specify bandwidth limits on specific days, specific hours, or both, listed most specific to least specific by using the syntax <code>[[Su Mo Tu We Th Fr Sa]][HH:00-HH:00][Kb/s],[ &lt;next_entry&gt;],[...]</code></p> <hr/> <p><b>Note:</b> If no bandwidth schedule is specified, all available bandwidth is used at all times.</p> <hr/> <p>Example:</p> <p>To set up an interconnect between Data Mover <code>server_3</code> on system <code>cs110</code> and its peer Data Mover <code>server_2</code> on system <code>cs100</code>, with a bandwidth schedule for weekdays from 7 A.M. to 4 P.M. at a bandwidth of 2000 Kb/s, at the destination site, type:</p> <pre>\$ nas_cel -interconnect -create cs110_s3s2 -source_server server_3   -destination_system cs100 -destination_server server_2 -source_interfaces   ip=192.168.52.146,ip=192.168.53.147,ip=192.168.168.148 -destination_interfaces   ip=192.168.52.136,ip=192.168.53.137,ip=192.168.168.138 -bandwidth MoTuWeThFr07:00   -16:00/2000</pre> <hr/> <p><b>Note:</b> Use the <code>nas_cel -list</code> command to learn the VNX system name. If the name is truncated or you find it difficult to extract from the output, display the full name clearly by using the <code>nas_cel -info</code> command with the VNX system ID.</p>
Output
<pre>operation in progress (not interruptible)... id = 20004 name = cs110_s3s2 source_server = server_3 source_interfaces = 192.168.52.146, 192.168.53.147 192.168.168.148 destination_system = cs100 destination_server = server_2 destination_interfaces = 192.168.52.136, 192.168.53.137, 192.168.168.138 bandwidth schedule = MoTuWeThFr07:00-16:00/2000 crc enabled = yes number of configured replications = 0 number of replications in transfer = 0 status = The interconnect is OK.</pre>

## Validate interconnect information

At both source and destination sites, check whether the interconnect is established between a given Data Mover and its peer:

- To list the interconnects available on the local VNX system and view the interconnect name and ID, use this command syntax:

```
$ nas_cel -interconnect -list
```

Output:

id	name	source_server	destination_system	destination_server
20001	loopback	server_2	cs100	server_2
20004	172143to17343	server_2	cs100	server_2
20005	cs100_s2s3	server_2	cs110	server_3
20006	ics2148s2	server_2	cs100	server_2

**Note:** All loopback interconnects display "loopback" for the interconnect name. Use the interconnect ID to specify a specific loopback interconnect.

2. To verify that both sides of the interconnect can communicate with each other, at each site, use this command syntax:

From either side:

```
$ nas_cel -interconnect -validate {<name>|id=<interconnectId>}
```

where:

<name> = name of the interconnect to validate (obtained from step 1)

<interconnectId> = ID of the interconnect to validate

Example:

To validate interconnect cs100\_s2s3 on server\_2, type:

```
$ nas_cel -interconnect -validate id=20005
```

Output:

```
validating...ok
```

**Note:** Validation fails if the peer interconnect is not set up, or for any interconnect that does not have interfaces from the same protocol in both source and destination lists. For example, if the source interface list includes an IPv6 address, the destination interface list must also include an IPv6 address.

## Configuring a file system replication session

You create a replication session for every file system you want to replicate. When you create a session, the system performs the following:

- ◆ Verifies whether source and destination Data Movers can communicate with each other
- ◆ Starts replication
- ◆ Begins tracking all changes made to the source file system

You set your replication policies when you establish the replication relationship. [Planning considerations on page 57](#) describes how to control replication sessions with replication policies.

---

**Note:** To create a session for a one-time copy of a file system, follow the procedures in [Chapter 8](#).

---

The tasks to create an ongoing replication session for a file system are:

- ◆ [Prerequisites on page 80](#)
- ◆ [Verify destination storage on page 81](#)
- ◆ [Validate Data Mover communication on page 81](#)
- ◆ [Create a file system replication session on page 82](#)
- ◆ [\(Optional\) Verify file system replication on page 85](#)

## Prerequisites

- ◆ You should have a functioning NFS or CIFS environment before you use VNX Replicator.
- ◆ If you are using VDMs in a CIFS environment, ensure that the VDM where the file system is located is replicated first.
- ◆ Verify that both the source and destination file systems have the same FLR type, either off, Enterprise, or Compliance. Also, when creating a file system replication, you cannot use an FLR-C-enabled destination file system that contains protected files.
- ◆ If you are using file systems enabled for data deduplication, verify that all destination file systems support VNX data deduplication.
- ◆ The procedures in this section assume the following:
  - The source file system is created and mounted as read/write on a Data Mover.
  - The destination file system may or may not be created. You can use an existing destination file system, especially for remote replications where the remote destination file system has the same name as the source.
  - The Data Mover interconnect is created.

If you want to choose a specific interface name or IP address for the replication session, specify it when you create the replication. You can specify a source and destination interface or allow the system to select it for you. If you allow the system to select it, the system will automatically verify that the source and destination interfaces can communicate. If you specify particular source and destination interfaces, there is no check to verify communication.

If you replicate a quotas-enabled file system from a system that runs on VNX for file version 6.0.41.0 or later to a destination system that runs on a version of VNX for file earlier than 6.0.41.0, the destination file system can become unusable.

Any future changes to this information requires stopping and starting replication, as detailed in [Stop a replication session on page 134](#) and [Start a replication session on page 136](#).

### Sample information

[Table 7 on page 80](#) lists information about the source and destination sites, Data Mover, and interconnects used in the examples in this section.

**Table 7. Sample information**

Site	VNX system name	Data Mover	Interconnect	Interfaces
Source	cs100	server_2	cs100_s2s3	192.168.52.136 192.168.53.137 192.168.168.138
Destination	cs110	server_3	cs110_s3s2	192.168.52.146 192.168.53.147 192.168.168.148



## Verify destination storage

There must be enough available storage on the destination side to create the destination file system.

Action
<p>To verify whether there is available storage to create the destination file system, use this command syntax:</p> <pre>\$ nas_pool -size &lt;name&gt;</pre> <p>where:</p> <p>&lt;name&gt; = name of the storage pool</p> <hr/> <p><b>Note:</b> Use the <code>nas_pool -list</code> command to obtain the name or ID of the storage pool.</p> <hr/> <p>Example:</p> <p>To verify the size of storage pool <code>clar_r1</code>, type:</p> <pre>\$ nas_pool -size clar_r1</pre>
Output
<pre>id           = 3 name        = clar_r1 used_mb     = 11485 avail_mb    = 2008967 total_mb    = 2020452 potential_mb = 0</pre>

## Validate Data Mover communication

Before you create the replication session, verify that authentication is configured properly between the Data Mover pair and validate all the combinations between source and destination IP addresses/interface names.

Action
<p>To verify Data Mover communication, use this command syntax:</p> <pre>\$ nas_cel -interconnect -validate {&lt;name&gt; id=&lt;interconnectId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the interconnect to validate</p> <p>&lt;interconnectId&gt; = ID of the interconnect to validate</p> <p>Example:</p> <p>To verify that authentication is configured properly for interconnect <code>cs100_s2s3</code>, type:</p> <pre>\$ nas_cel -interconnect -validate cs100_s2s3</pre>

Action
<b>Note:</b> To obtain the interconnect ID, use the <code>nas_cel -interconnect -list</code> command.
Output
validating...ok

## Create a file system replication session

To create a file system replication session, see the examples below.

Action
<p>To create a file system replication session that uses a storage pool, a system-selected local interface supporting the lowest number of sessions, and specifies a time-out-of-sync value, use this command syntax:</p> <pre>\$ nas_replicate -create &lt;name&gt; -source -fs {&lt;fsName&gt; id=&lt;fsId&gt;} -destination -pool {id=&lt;dstStoragePoolId&gt; &lt;dstStoragePool&gt;} -interconnect {&lt;name&gt; id=&lt;interConnectId&gt;} -max_time_out_of_sync &lt;maxTimeOutOfSync&gt;</pre>
<p><b>Note:</b> If you are using an existing file system, use the <code>-destination -fs</code> option and specify the file system name or ID.</p>
<p>where:</p> <p><code>&lt;name&gt;</code> = name of the replication session.</p> <p><code>&lt;fsName&gt;</code> = name of the existing source file system to replicate. The source and destination file systems must have the same FLR type, either off, Enterprise, or Compliance.</p> <p><code>&lt;fsId&gt;</code> = ID of the existing source file system to replicate.</p> <p><code>&lt;dstStoragePoolId&gt;</code> = ID of the storage pool to use to create the destination file system.</p> <p><code>&lt;dstStoragePool&gt;</code> = name of the storage pool to use to create the destination file system.</p> <p><code>&lt;name&gt;</code> = name of the local source side Data Mover interconnect.</p> <p><code>&lt;interConnectID&gt;</code> = ID of the local source-side Data Mover interconnect.</p> <p><code>&lt;maxTimeOutOfSync&gt;</code> = number of minutes that the source and destination can be out of synchronization before an update occurs.</p>
<p>Example:</p> <p>To create remote replication session <code>rep1</code> for source file system <code>src_ufs1</code> to a destination by using destination storage pool <code>clar_r1</code>, and specifying a refresh update of 15 minutes, type:</p> <pre>\$ nas_replicate -create rep1 -source -fs src_ufs1 -destination -pool clar_r1 -interconnect cs100_s2s3 -max_time_out_of_sync 15</pre>
<p><b>Note:</b> The system selects the interface if you do not specify one.</p>

**Output**

OK

**Verification**

To verify that the replication session was created, type:

```
$ nas_replicate -list
```

Output:

Name	type	Local Mover	Interconnect	Celerra	Status
rep1	filesystem	server_2	-->cs100_s2s3	cs_100	OK

The next example creates a replication session that uses specific source and destination interfaces and a manual update policy.

**Action**

To create a file system replication that has no existing destination file system, uses specific IP addresses, and a manual update policy, use this command syntax:

```
$ nas_replicate -create <name> -source -fs <fsName> -destination
-pool {id=<dstStoragePoolId>|<dstStoragePool>}
-interconnect {<name>|id=<interConnectId>}]
-source_interface {<name_service_interface_name>|ip=<ipaddr>}]
-destination_interface {<name_service_interface_name>|ip=<ipaddr>}]
-manual_refresh
```

where:

<name> = name of the replication session.

<fsName> = name of existing source file system to replicate. The source and destination file systems must have the same FLR type, either off, Enterprise, or Compliance.

<dstStoragePoolId> = ID of the storage pool to use to create the destination file system.

<dstStoragePool> = name of the storage pool to use to create the destination file system.

<name> = name of the local source side Data Mover interconnect.

<interConnectId> = ID of the local source side Data Mover interconnect.

<name\_service\_interface\_name> = interface name to use first for the source interface interconnect and then the destination interface. This network interface name must resolve to a single IP address (for example, by a DNS server.)

<ipaddr> = IP address (IPv4 or IPv6) to use first for the source interface interconnect and then the destination interface interconnect.

**Note:** Ensure that the source and destination interface lists use the same IPv4/IPv6 protocol. An IPv4 interface cannot connect to an IPv6 interface and vice versa. Both sides of the connection must use the same protocol.

Example:

To create a file system replication session named rep2 to replicate the source file system src\_ufs1 specifying certain IP addresses to use as interfaces and a manual update policy, type:

Action
<pre>\$ nas_replicate -create rep2 -source -fs src_ufs1 -destination -pool=dst_pool1 -interconnect cs100_s2s3 -source_interface ip=192.168.52.136 -destination_interface ip=192.168.52.146 -manual_refresh</pre>
Output
OK

The next example creates a replication session that uses an existing destination file system and the default `max_time_out_of_sync` value of 10 minutes.

**Note:** When you create a replication session by using an existing, populated destination file system, it uses more of the destination file system SavVol space. Therefore, if the existing data on the destination is not needed, delete the destination file system and recreate the session using the `-pool` option. The SavVol space for a file system can easily exceed the total space of the primary file system. This is because the SavVol space can be 20 percent of the total storage space assigned to the system.

Action
<p>To create a file system replication that has an existing destination file system, use this command syntax:</p> <pre>\$ nas_replicate -create &lt;name&gt; -source -fs &lt;fsName&gt; -destination -fs {id=&lt;dstFsId&gt; &lt;existing_dstFsName&gt;} -interconnect {&lt;name&gt; id=&lt;interConnectId&gt;}</pre> <p>where:</p> <p><code>&lt;name&gt;</code> = name of the replication session.</p> <p><code>&lt;fsName&gt;</code> = name of existing source file system to replicate. The source and destination file systems must have the same FLR type, either off, Enterprise, or Compliance.</p> <p><code>&lt;dstFsId&gt;</code> = ID of the existing file system to use on the destination. The source and destination file systems must have the same FLR type, either off, Enterprise, or Compliance. You cannot use an FLR-C-enabled destination file system that contains protected files.</p> <p><code>&lt;existing_dstFsName&gt;</code> = name of the existing file system to use on the destination. The source and destination file systems must have the same FLR type, either off, Enterprise, or Compliance. You cannot use an FLR-C-enabled destination file system that contains protected files.</p> <p><code>&lt;name&gt;</code> = name of the local source side Data Mover interconnect.</p> <p><code>&lt;interConnectId&gt;</code> = ID of the local source side Data Mover interconnect.</p> <p><b>Note:</b> If you do not enter a <code>max_time_out_of_sync</code> value or specify manual refresh, the system will default to 10 minutes for a file system session.</p> <p>Example:</p> <p>To create a file system replication session named <code>rep2</code> to replicate the source file system <code>src_ufs1</code> and specifying an existing destination file system, type:</p> <pre>\$ nas_replicate -create rep2 -source -fs src_ufs1 -destination -fs src_ufs1_dst -interconnect cs100_s2s3</pre>

**Output**

OK

## (Optional) Verify file system replication

Use the `nas_replicate -info` command at any time to view detailed information about a replication session on a VNX system. [Get information about replication sessions on page 128](#) provides more details about this command.

**Action**

To check the status of a specific replication session, use this command syntax:

```
$ nas_replicate -info {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

Example:

To display the information about replication session `rep2`, which includes status information, type:

```
$ nas_replicate -info rep2
```

**Note:** Use the `nas_task -list` command to list all local tasks that are in progress, or completed tasks that have not been deleted.

## Output

```

ID =
1106_APM00063501064_0024_2791_APM00063501064_0024
Name = rep2
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time =
Type = filesystem
Celerra Network Server = cs100
Dart Interconnect = cs100_s2s3
Peer Dart Interconnect = cs110_s3s2
Replication Role = source
Source Filesystem = src_ufs1
Source Data Mover = server_2
Source Interface = 192.168.52.136
Source Control Port = 0
Destination Filesystem = src_ufs1_dst
Destination Data Mover = server_3
Destination Interface = 192.168.52.146
Destination Control Port = 5081
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 20
Next Transfer Size (KB) = 4680
Current Transfer Size (KB) = 4680
Current Transfer Remain (KB) = 4680
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 0
Current Read Rate (KB/s) = 0
Current Write Rate (KB/s) = 0
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 0
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

```

## Configuring a VDM replication session

The VDM replication process consists of two main tasks: First, you create the session to replicate the VDM and then you create a session to replicate the file system mounted to the VDM. [Replication source objects on page 21](#) provides more information on replicating VDM source objects.

The tasks to create a VDM replication session are:

- ◆ [Prerequisites on page 88](#)
- ◆ [Verify the source VDM on page 88](#)
- ◆ [Verify available destination storage on page 88](#)
- ◆ [Validate Data Mover communication on page 89](#)
- ◆ [Create VDM replication session on page 90](#)
- ◆ [Replicate the file system mounted to the VDM on page 91](#)
- ◆ [\(Optional\) Verify VDM replication on page 92](#)

## Prerequisites

Before you create a VDM replication session note the following:

- ◆ Review [Appendix A](#) to learn more about the CIFS replication environment.
- ◆ The source and destination side interface names must be the same for CIFS servers transition. You may need to change the interface name created by using Unisphere (by default it uses IP address) by creating a new interface name with the `server_ifconfig` command.
- ◆ The source and destination side mount points must be the same for the share names to resolve correctly. This will ensure the CIFS share can recognize the full path to the share directory and users will be able to access the replicated data after failover.
- ◆ When the source replicated file system is mounted on a VDM, the destination file system should also be mounted on the VDM that is replicated from the same source VDM.
- ◆ For the NFS endpoint of a replicated VDM to work correctly on the destination side, the Operating Environments of the source and destination sides must be version 7.0.50.0 or later.
- ◆ The local groups in the source VDM are replicated to the destination side in order to have complete access control lists (ACLs) on the destination file system. *Configuring and Managing CIFS on VNX* provides more information.

## Verify the source VDM

The source VDM to be replicated must be in the loaded read/write or mounted read-only state.

Action					
To display a list of all VDMs in the Data Mover server table, use this command syntax:					
<code>\$ nas_server -list -vdm</code>					
Output					
id	acl	server	mountedfs	rootfs	name
1	0	1		991	vdm1
2	0	2		993	vdm2
3	0	2		1000	vdm_1G
4	0	2		1016	vdm3

## Verify available destination storage

The destination VDM will be created automatically, the same size as the source and read-only as long as the specified storage is available.



Follow the procedure in [Verify the source VDM on page 88](#) to verify a destination VDM exists.

Action
<p>To verify the storage pool to use to create the destination VDM, use this command syntax:</p> <pre>\$ nas_pool -size &lt;name&gt;</pre> <p>where:</p> <p>&lt;name&gt; = name of the storage pool</p> <p>Use the nas_pool -list command to obtain the name or ID of the storage pool.</p> <p>Example:</p> <p>To verify the size of storage pool vdm_pool, type:</p> <pre>\$ nas_pool -size vdm_pool</pre>
Output
<pre>id          = 17 name        = vdm_pool used_mb     = 25404 avail_mb    = 508859 total_mb    = 534263 potential_mb = 0</pre>

## Validate Data Mover communication

Before you create the replication session, verify that authentication is configured properly between the Data Mover pair and validate all the combinations between source and destination IP addresses/interface names.

Action
<p>To verify Data Mover communication, use this command syntax:</p> <pre>\$ nas_cel -interconnect -validate {&lt;name&gt; id=&lt;interconnectId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the interconnect to validate</p> <p>&lt;interconnectId&gt; = ID of the interconnect to validate</p> <p>Example:</p> <p>To verify that authentication is configured properly for interconnect cs100_s2s3, type:</p> <pre>\$ nas_cel -interconnect -validate cs100_s2s3</pre>
<p><b>Note:</b> To obtain the interconnect ID, use the nas_cel -interconnect -list command.</p>

**Output**

```
validating...ok
```

## Create VDM replication session

The VDM replication session replicates information contained in the root file system of a VDM. It produces a point-in-time copy of a VDM that re-creates the CIFS environment at a destination. It does not replicate the file systems mounted to the VDM.

A given source VDM can support up to four replication sessions.

**Action**

To create a VDM replication session, use this command syntax:

```
$ nas_replicate -create <name> -source -vdm <vdmName> -destination
-vdm <existing_dstVdmName> -interconnect {<name>|id=<interConnectId>}
[{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}]
```

where:

<name> = name of the VDM replication session.

<vdmName> = name of the source VDM to replicate. VDM must be in the loaded or mounted state and can already be the source or destination VDM of another replication.

<existing\_dstVdmName> = name of the existing VDM on the destination. VDM must be in the mounted state. VDM can be the source of another replication but cannot be the destination of another replication.

<name> = name of the local source side of the established Data Mover interconnect to use for this replication session.

<interConnectId> = ID of the local source side of the established Data Mover interconnect to use for this replication session.

<maxTimeOutOfSync> = number of minutes that the source and destination can be out of synchronization before an update occurs.

**Note:** If you specify a pool instead of an existing destination VDM, the pool is used to create the destination VDM automatically as read-only and uses the same name and size as the source VDM.

Example:

To create VDM replication session rep\_vdm1 by using the source vdm1 and with existing destination VDM dst\_vdm with a refresh rate of 10 minutes, type:

```
$ nas_replicate -create rep_vdm1 -source -vdm vdm1 -destination -vdm dst_vdm
-interconnect cs100_s2s3 -max_time_out_of_sync 10
```

**Output**

```
OK
```

**Verification**

To verify that rep\_vdm1 was created, type:

**Verification**

```
$ nas_replicate -list
```

Name	type	Local Mover	Interconnect	Celerra	Status
rep_vdm1	vdm	server_2	-->cs100_s2s3	cs100	OK

## Replicate the file system mounted to the VDM

After you have created the session to replicate the VDM, create a session to replicate the file system mounted to the VDM.

**Action**

To create a file system replication session mounted to a VDM, use the following syntax:

```
$ nas_replicate -create <name> -source -fs {<fsName>|id=<fsid>}
-destination -pool {id=<dstStoragePoolId>|<dstStoragePool>}
-vdm {<existing_dstVdmName>|id=<dstVdmId>}
-interconnect {<name>|id=<interConnectId>}
[{-max_time_out_of_sync <maxTimeOutOfSync>|-manual_refresh}]
```

where:

<name> = name of the file system replication session

<fsName> = name of the source file system mounted to the VDM

<fsid> = ID of the source file system mounted to the VDM

<dstStoragePoolId> = ID of the storage pool to use to create the destination file system

<dstStoragePool> = name of the storage pool to use to create the destination file system

<existing\_dstVdmName> = name of an existing destination VDM on which to mount the destination file system

<dstVdmId> = ID of the destination VDM on which to mount the destination file system

<interConnectId> = ID of the local side of an established Data Mover interconnect to use for this replication session

<name> = name of the local side of an established Data Mover interconnect to use for this replication session

<maxTimeOutOfSync> = number of minutes that the source and destination can be out of synchronization before an update occurs

Example:

To create a file system replication session mounted to an existing VDM, type:

```
$ nas_replicate -create vdm_v2 -source -fs src_fs_vdm -destination -pool id=3 -vdm
vdm_2 -interconnect cs100_s2s3
```

**Note:** Use the -vdm option used to indicate that the file system will be mounted on a VDM.

**Output**

```
OK
```

## (Optional) Verify VDM replication

Use the `nas_replicate -info` command at any time to view detailed information about a replication session on a VNX system. [Get information about replication sessions on page 128](#) provides more details about this command.

### Action

To verify the status of a specific replication session, use this command syntax:

```
$ nas_replicate -info {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

Example:

To display the status of the VDM replication session, type:

```
$ nas_replicate -info vdm_v2
```

**Note:** You can also use the `nas_task -list` command to list all local tasks that are in progress, or completed tasks that have not been deleted.

**Output**

```

ID = 10867_APM00063303672_0000_25708_APM00063303672_0000
Name = vdm_v2
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time =
Type = vdm
Celerra Network Server = eng17343
Dart Interconnect = loopback
Peer Dart Interconnect = loopback
Replication Role = loopback
Source VDM = srcvdm1_replica1
Source Data Mover = server_2
Source Interface = 127.0.0.1
Source Control Port = 0
Source Current Data Port = 0
Destination VDM = srcvdm1
Destination Data Mover = server_2
Destination Interface = 127.0.0.1
Destination Control Port = 5081
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 2
Next Transfer Size (KB) = 0
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 4000
Current Read Rate (KB/s) = 40960
Current Write Rate (KB/s) = 3461
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 3585
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

```



## Configuring a one-time copy session

This section lists the tasks to create a session for a one-time copy of a source file system mounted read-only, a file system checkpoint mounted read-only, or a file system writable checkpoint mounted read-only to the same or different VNX system.

The tasks to create a one-time file system copy session are:

- ◆ [Prerequisites on page 96](#)
- ◆ [Verify the source file system on page 96](#)
- ◆ [Verify destination file system or storage on page 97](#)
- ◆ [Validate Data Mover communication on page 99](#)
- ◆ [Create a file system copy session on page 99](#)
- ◆ [\(Optional\) Monitor file system copy on page 102](#)

---

## Prerequisites

Using `nas_copy` constitutes a replication session. The command copies source data to the destination and then deletes the session and all internal checkpoints created on the source and destination. [How one-time file system copy sessions work on page 42](#) explains how one-time copy sessions work.

Multiple copy sessions can run from the same source. Each copy session counts towards the four replication sessions limit and the number of sessions limit per Data Mover.

Before creating a one-time copy session, note the following:

- ◆ The source read-only file system or file system checkpoint must already exist.
- ◆ The destination file system or the storage pool required to create the destination file system must exist.
- ◆ The destination file system must be the same size as the source and is read-only accessible.
- ◆ You cannot create a copy session unless both the source and destination file systems have the same FLR type, either off, Enterprise, or Compliance. Also, when creating a file system replication, you cannot use an FLR-C-enabled destination file system that contains protected files.
- ◆ The Data Mover interconnect to use for the session must be set up.

### `nas_copy` update options

The `nas_copy` update options work as follows:

- ◆ **Overwrite destination** — For an existing destination file system, discards any changes made to the destination file system and restores it from the established common base (differential copy). If you do not specify this option and an existing destination file system contains different content, the command returns an error.
- ◆ **Refresh** — Updates a destination checkpoint that has the same name as the copied checkpoint. Use this option when copying an existing checkpoint, instead of creating a new checkpoint at the destination after the copy. If no checkpoint exists with the same name, the command returns an error.
- ◆ **Full copy** — For an existing file system, if a common base checkpoint exists, performs a full copy of the source checkpoint to the destination instead of a differential copy.

---

**Note:** You must explicitly specify the full copy option when starting a copy session and no established common base exists. If full copy is not specified, the copy session will fail.

---

---

## Verify the source file system

Before you create a copy session, verify that the source file system exists and that it is mounted read-only.



1. To verify that the source file system exists on the Data Mover, type:

```
$ nas_fs -list
```

Output:

id	inuse	type	acl	volume	name	server
58	y	1	0	204	fs1	2
60	y	1	0	208	fs2	1
62	y	1	0	214	fs4	2
90	y	1	0	252	dxfps	1
91	y	1	0	254	fs111	1
94	n	7	0	261	fs111_ckpt	1
98	y	1	0	272	fs113	2
101	y	15	0	261	fs111_ckpt1_writeab	1

2. To verify that the source file system is mounted read-only, use this command syntax:

```
$ server_mount <movername>
```

where:

<movername> = name of the Data Mover where the source file system is mounted

Example:

To verify that source file system fs113 is mounted read-only, type:

```
$ server_mount server_2
```

Output:

```
server_2 :
root_fs_2 on / udfs,perm,rw
root_fs_common on /.etc_common udfs,perm,ro
fs2 on /fs2 udfs,perm,rw
ca1 on /ca1 udfs,perm,rw
ca2 on /ca2 udfs,perm,rw
ca3 on /ca3 udfs,perm,ro
ca3_replica1 on /ca3_replica1 udfs,perm,ro
root_fs_vdm_vdm_mkt on /root_vdm_1/.etc udfs,perm,rw
ca3_replica2 on /ca3_replica2 udfs,perm,ro
dxfps on /dxfps udfs,perm,rw
fs111 on /fs111 udfs,perm,rw
fs113 on /fs113 udfs,perm,ro
```

## Verify destination file system or storage

When there is no existing destination file system, you must verify that there is a storage pool available and that it includes enough storage for the copy. If you are copying to an existing destination file system, verify that the file system exists, is the same size as the source, and that it is mounted read-only.

1. If you are copying to an existing destination file system, verify that the destination file system exists on the Data Mover, type:

```
$ nas_fs -list
```

Output:

id	inuse	type	acl	volume	name	server
94	y	1	0	254	fs1_replica1	2
96	y	7	0	257	root_rep_ckpt_94_10	2
97	y	7	0	257	root_rep_ckpt_94_10	2
98	y	1	0	259	fs113	2

**Note:** The output shows that file system fs113 exists on destination Data Mover server \_2.

- To verify that the destination file system is mounted read-only, use this command syntax:

```
$ server_mount <movername>
```

where:

<movername> = name of the Data Mover where the destination file system is mounted

Example:

To verify that destination file system fs113 is mounted read-only, from the destination side, type:

```
$ server_mount server_2
```

Output:

```
server_2 :
root_fs_2 on / uxfs,perm,rw,<unmounted>
root_fs_common on /.etc_common uxfs,perm,ro,<unmounted>
fs3 on /fs3 uxfs,perm,rw,<unmounted>
smm4_replica1 on /smm4_replica1 uxfs,perm,ro,<unmounted>
smm4-silver on /smm4-silver ckpt,perm,ro,<unmounted>
smm4_cascade on /smm4_cascade rawfs,perm,ro,<unmounted>
smm1 on /smm1 uxfs,perm,ro,<unmounted>
dxsfs on /dxsfs uxfs,perm,rw,<unmounted>
fs113 on /fs113 uxfs,perm,ro,
```

- If there is no existing destination file system, verify that there is a storage pool and that it includes enough storage for the copy. To list the available storage pools for the destination VNX system, from the destination side, type:

```
$ nas_pool -list
```

Output:

id	inuse	acl	name	storage_system
2	y	0	clar_r1	N/A
3	y	0	clar_r5_performance	FCNTR074200038

**Note:** The output shows the storage pool IDs. Use the pool ID to view the available storage for the pool in step 4.

- To view the available storage for a specific pool, use this command syntax:

```
$ nas_pool -size {<name>|id=<id>|-all}
```

Example:

To view the available storage for pool 3, type:

```
$ nas_pool -size id=3
```

Output:

```
id           = 3
name         = clar_r5_performance
used_mb      = 44114
avail_mb     = 1988626
total_mb     = 2032740
potential_mb = 0
```

**Note:** The output shows the available storage in MBs (avail\_mb).

## Validate Data Mover communication

Before you create the replication session, verify that authentication is configured properly between the Data Mover pair and validate all the combinations between source and destination IP addresses/interface names.

Action
<p>To verify Data Mover communication, use this command syntax:</p> <pre>\$ nas_cel -interconnect -validate {&lt;name&gt; id=&lt;interconnectId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the interconnect to validate</p> <p>&lt;interconnectId&gt; = ID of the interconnect to validate</p> <p>Example:</p> <p>To verify that authentication is configured properly for interconnect cs100_s2s3, type:</p> <pre>\$ nas_cel -interconnect -validate cs100_s2s3</pre> <p><b>Note:</b> To obtain the interconnect ID, use the nas_cel -interconnect -list command.</p>
Output
<pre>validating...ok</pre>

## Create a file system copy session

Execute the nas\_copy command from the Control Station on the source side only.

**Action**

To create a one-time copy session for a source read-only file system with no existing destination, use this command syntax:

```
$ nas_copy -name <sessionName> -source -fs {<name>|id=<fsId>}
-destination -pool {id=<dstStoragePoolId>|<dstStoragePool>}
-interconnect {<name>|id=<interConnectId>}
```

where:

<sessionName> = name of the one-time copy session

<name> = name of the existing source read-only file system to be copied

<fsId> = ID of the existing source read-only file system to be copied

<dstStoragePoolId> = ID of the storage pool to use to create the destination file system

<dstStoragePool> = name of the storage pool to use to create the destination file system

<name> = name of the interconnect configured between source and destination Data Movers

<interconnectId> = ID of the interconnect configured between source and destination (peer) Data Movers

Example:

To create a one-time copy session ca3\_copy for source file system fs113 that has no existing destination file system, from the source side, type:

```
$ nas_copy -name ca3_copy -source -fs fs113 -destination -pool id=3 -interconnect
id=20001
```

**Output**

OK

**Notes**

Once started, you can monitor the session by using the `nas_task` command. [Chapter 11](#) provides more information.

[Return codes for nas\\_copy on page 193](#) lists the return codes generated by the `nas_copy` command, and can be used for error checking purposes.

Depending on the size of the data on the source, this command may take some time to complete.

This is an example of a file system copy with an existing destination file system.

**Action**

To create a one-time copy session for a source read-only file system when the file system already exists on the destination, use this command syntax:

```
$ nas_copy -name <sessionName> -source -fs {<name>|id=<fsId>}
-destination -fs {id=<dstFsId>|<existing_dstFsName>}
-interconnect {<name>|id=<interConnectId>} -overwrite_destination -full_copy
```

where:

<sessionName> = name of the one-time copy session

<name> = name of the existing source read-only file system to be copied

Action
<p>&lt;fsId&gt; = ID of the existing source read-only file system to be copied</p> <p>&lt;dstFsId&gt; = ID of the existing destination file system</p> <p>&lt;existing_dstFsName&gt; = name of the existing destination file system</p> <p>&lt;name&gt; = name of the interconnect configured between source and destination Data Movers</p> <p>&lt;interConnectId&gt; = ID of the interconnect configured between source and destination (peer) Data Movers</p> <p>Example:</p> <p>To create a copy session copy_src_fs for source file system src_fs when the file system dest_fs already exists on the destination, type:</p> <pre>\$ nas_copy -name copy_src_fs -source -fs src_fs -destination -fs dest_fs -interconnect id=30002 -overwrite_destination -full_copy</pre>
Output
OK

Notes
<p>This will overwrite the contents on the destination.</p> <p>To verify that the file system was created on the destination, use the nas_fs -list command.</p> <p>Once started, you can monitor the session by using the nas_task command. <a href="#">Chapter 11</a> provides more information.</p> <p><a href="#">Return codes for nas_copy on page 193</a> lists the return codes generated by the nas_copy command, and can be used for error checking purposes.</p>

This is an example of creating a copy session for a file system checkpoint.

Action
<p>To create a one-time copy session for a file system checkpoint, use this command syntax:</p> <pre>\$ nas_copy -name &lt;sessionName&gt; -source {-ckpt &lt;ckptName&gt; id=&lt;ckptId&gt;} -destination -pool {id=&lt;dstStoragePoolId&gt; &lt;dstStoragePool&gt;} -interconnect {&lt;name&gt; id=&lt;interConnectId&gt;}</pre> <p>where:</p> <p>&lt;sessionName&gt; = name of the one-time copy session</p> <p>&lt;ckptName&gt; = name of the existing source read-only file system checkpoint to be copied</p> <p>&lt;ckptId&gt; = ID of the existing source read-only file system checkpoint to be copied</p> <p>&lt;dstStoragePoolId&gt; = ID of the storage pool to use to create the destination file system</p> <p>&lt;dstStoragePool&gt; = name of the storage pool to use to create the destination file system</p> <p>&lt;name&gt; = name of the interconnect configured between source and destination Data Movers</p> <p>&lt;interConnectId&gt; = ID of the interconnect configured between source and destination Data Movers</p> <p>Example:</p>

### Action

To create one-time copy session `copyrep6_copy` for file system checkpoint `rep5_ckpt1`, type:

```
$ nas_copy -name copyrep6_copy -source -ckpt rep5_ckpt1 -destination -pool id=3  
-interconnect id=20003
```

### Output

OK

### Notes

To verify that `rep5_ckpt1` was created on the destination, use the `nas_fs -list` command.

Once started, you can monitor the session by using the `nas_task` command. [Chapter 11](#) provides more information.

[Return codes for nas\\_copy on page 193](#) lists the return codes generated by the `nas_copy` command, and can be used for error checking purposes.

## (Optional) Monitor file system copy

Once started, you can monitor the copy session by using the `nas_task` command. This command is used to get information about long-running tasks, such as a replication copy task. You can display the status of a specific copy session or all copy sessions. [Chapter 11](#) provides more information.

### Action

To display detailed information about a task, use this command syntax:

```
$ nas_task -info {-all | <taskId>}
```

where:

`<taskId>` = ID of the replication copy session

Use the `nas_task -list` command to obtain the task ID for the copy session.

Example:

To display detailed information about `rep5_copy`, with task ID 9730, type:

```
$ nas_task -info 9730
```

**Output**

```
Task Id = 9730
Celerra Network Server = Local System
Task State = Running
Current Activity = Info 26045317613: Transfer done
Movers =
Percent Complete = 94
Description = Create Replication rep5_copy.
Originator = nasadmin@10.240.4.99
Start Time = Thu Jan 24 11:22:37 EST 2008
Estimated End Time = Thu Jan 24 11:21:41 EST 2008
Schedule = n/a
```





## Configuring advanced replications

With VNX Replicator, you can configure one-to-many and cascade replication configurations. You can also replicate a file system mounted on a Nested Mount file system mount point

The tasks to configure these advanced replication configurations are:

- ◆ [Configure a one-to-many replication on page 106](#)
- ◆ [Configure a cascade replication on page 106](#)
- ◆ [Configure a one-to-many replication with common base checkpoints on page 107](#)
- ◆ [Configure a cascading replication with common base checkpoints on page 115](#)
- ◆ [Configure the replication of a file system mounted on an Nested Mount File System mountpoint on page 124](#)

---

## Configure a one-to-many replication

To take advantage of a one-to-many configuration, where the source object is used in more than one replication session, create multiple sessions on the source and associate the source for each session with up to four different destinations. This configuration is actually a one-to-one created up to four times. All source object types can be used in a one-to-many configuration.

[One-to-many replication configurations on page 27](#) explains this configuration in more detail.

1. Create a replication session. Follow the appropriate procedure for the type of replication session for which you want to create the configuration:
  - [Chapter 6](#) for a file system replication session
  - [Chapter 7](#) for VDM replication
  - [Chapter 8](#) for a one-time copy session
2. Repeat step 1 up to three more times by using the same source object. If it is a remote replication session, ensure to specify a remote interconnect.

---

## Configure a cascade replication

To take advantage of a cascade configuration where one destination also serves as the source for another replication session, create a replication session on each source in the path. The first session runs replication from the Source to Destination 1, and the second session runs replication from Destination 1, serving as the source, to Destination 2. Replication sessions in a cascade configuration are supported up to two network hops. All source object types can be used in a cascade configuration.

[Cascade replication configurations on page 31](#) explains this configuration in more detail.

1. Create a session for ongoing replication. Follow the appropriate procedure for the type of replication session for which you want to create a cascade configuration:
  - [Chapter 6](#) for a file system replication session
  - [Chapter 7](#) for VDM replication
  - [Chapter 8](#) for a one-time copy session
2. Create a second session on the destination side and for the source object, select the name of the destination object replicated at destination 1 from step 1.

---

**Note:** Use `nas_replicate -info` to view the destination file system name.

---

---

## Configure a one-to-many replication with common base checkpoints

1. Create a user checkpoint for each file system by using the following command syntax:

```
# fs_ckpt <fs_name> -name <ckpt_name> -Create
```

where:

<fs\_name> = name of the file system

<ckpt\_name> = name of the user checkpoint created for the file system

---

**Note:** The `-name` command is optional. If you do not use the `-name` command to customize the names of the checkpoints, the system names the checkpoints. However, EMC recommends that you customize the names of the checkpoints for ease of management.

---

Example:

```
[root@cse17107 nasadmin]# fs_ckpt otm -name otm_ckptA -Create
```

Output:

```

operation in progress (not interruptible)...id          = 107
name           = otm
acl            = 0
in_use        = True
type          = uxfs
worm          = off
volume        = v245
pool          = clarata_archive
member_of     = root_avm_fs_group_10
rw_servers    = server_2
ro_servers    =
rw_vdms       =
ro_vdms       =
auto_ext      = no,thin=no
deduplication = Off
ckpts         = root_rep_ckpt_107_53052_2,root_rep_ckpt_107_53052_1,otm_ckptA
rep_sess      = 245_FNM00103800639_2007_254_FNM00103800639_2007(ckpts:
root_rep_ckpt_107_53052_1, root_rep_ckpt_107_53052_2)
stor_devs     = FNM00103800639-000F
disks         = d18
disk=d18      stor_dev=FNM00103800639-000F addr=c16t1111
server=server_2
disk=d18      stor_dev=FNM00103800639-000F addr=c0t1111
server=server_2
id            = 119
name          = otm_ckptA
acl           = 0
in_use        = True
type          = ckpt
worm          = off
volume        = vp252
pool          = clarata_archive
member_of     =
rw_servers    =
ro_servers    = server_2
rw_vdms       =
ro_vdms       =
checkpoint_of = otm Wed Aug 24 11:11:52 EDT 2011
deduplication = Off
used          = 8%
full(mark)    = 90%
stor_devs     = FNM00103800639-000F
disks         = d18
disk=d18      stor_dev=FNM00103800639-000F addr=c16t1111
server=server_2
disk=d18      stor_dev=FNM00103800639-000F addr=c0t1111
server=server_2

```

```
[root@cse17107 nasadmin]# fs_ckpt otm_replica1 -name otm_replica1_ckptB -Create
```

Output:

```

operation in progress (not interruptible)...id           = 113
name            = otm_replica1
acl             = 0
in_use         = True
type           = uxfs
worm           = off
volume        = v254
pool          = clarata_archive
member_of     = root_avm_fs_group_10
rw_servers=
ro_servers= server_3
rw_vdms      =
ro_vdms      =
auto_ext     = no,thin=no
deduplication = Off
ckpts        = root_rep_ckpt_113_53065_2,root_rep_ckpt_113_53065_1,
              root_rep_ckpt_113_53280_2,root_rep_ckpt_113_53280_1,
              otm_replica1_ckptB
rep_sess     = 245_FNM00103800639_2007_254_FNM00103800639_2007(ckpts:
root_rep_ckpt_113_53065_1, root_rep_ckpt_113_53065_2),
254_FNM00103800639_2007_129_FNM00103600044_2007(ckpts:
root_rep_ckpt_113_53280_1, root_rep_ckpt_113_53280_2)
stor_devs    = FNM00103800639-0009
disks        = d12
              disk=d12  stor_dev=FNM00103800639-0009  addr=c16t115
server=server_3
              disk=d12  stor_dev=FNM00103800639-0009  addr=c0t115
server=server_3

id           = 120
name        = otm_replica1_ckptB
acl         = 0
in_use     = True
type       = ckpt
worm       = off
volume    = vp257
pool      = clarata_archive
member_of =
rw_servers=
ro_servers= server_3
rw_vdms  =
ro_vdms  =
checkpt_of= otm_replica1 Wed Aug 24 11:13:02 EDT 2011
deduplication = Off
used         = 13%
full(mark)  = 90%
stor_devs   = FNM00103800639-000D
disks       = d16
              disk=d16  stor_dev=FNM00103800639-000D  addr=c16t119
server=server_3
              disk=d16  stor_dev=FNM00103800639-000D  addr=c0t119
server=server_3

```

**Note:** The following checkpoint creation is on the destination side. The name of the file system is the same as the source because of replication.

```
[nasadmin@ENG19105 ~]$ fs_ckpt otm_replica1 -name otm_ckptC -Create
```

Output:

```

operation in progress (not interruptible)...id          = 113
name           = otm
acl            = 0
in_use        = True
type          = uxfs
worm          = off
volume        = v129
pool          = clarsas_archive
member_of     = root_avm_fs_group_32
rw_servers=
ro_servers= server_2
rw_vdms       =
ro_vdms       =
auto_ext      = no,thin=no
deduplication = Off
ckpts         = root_rep_ckpt_113_9605_2,root_rep_ckpt_113_9605_1,otm_ckptC
rep_sess      = 254_FNM00103800639_2007_129_FNM00103600044_2007(ckpts:
root_rep_ckpt_113_9605_1, root_rep_ckpt_113_9605_2)
stor_devs     = FNM00103600044-0000
disks         = d8
  disk=d8     stor_dev=FNM00103600044-0000 addr=c16t110
server=server_2
  disk=d8     stor_dev=FNM00103600044-0000 addr=c0t110
server=server_2

id            = 117
name          = otm_ckptC
acl           = 0
in_use        = True
type          = ckpt
worm          = off
volume        = vp132
pool          = clarsas_archive
member_of     =
rw_servers=
ro_servers= server_2
rw_vdms       =
ro_vdms       =
ckpt_of= otm Wed Aug 24 11:13:55 EDT 2011
deduplication = Off
used          = 8%
full(mark)= 90%
stor_devs     = FNM00103600044-0000
disks         = d8
  disk=d8     stor_dev=FNM00103600044-0000 addr=c16t110
server=server_2
  disk=d8     stor_dev=FNM00103600044-0000 addr=c0t110
server=server_2

```

2. On the source VNX, refresh the local replication session from A to B with the user-created checkpoints by using the following command syntax:

```

$ nas_replicate -refresh <rep_session_name_AtoB> -source <ckpt_name_A>
-destination <ckpt_name_B >

```

where:

<rep\_session\_name\_AtoB> = name of the replication session between A and B

<ckpt\_name\_A> = name of the user checkpoint for source VNX A

<ckpt\_name\_B> = name of the user checkpoint for destination VNX B

After the refresh completes, the user-created checkpoints for VNX A and VNX B have a common baseline for the replication pair.

Example:

```
[root@cse17107 nasadmin]# nas_replicate -refresh 17107s2-17107s3 -source otm_ckptA
-destination otm_replica1_ckptB
```

Output:

OK

3. On the source VNX, refresh the remote replication session from A to C with the user-created checkpoints by using the following command syntax:

```
$ nas_replicate -refresh <rep_session_name_AtoC> -source <ckpt_name_A>
-destination <ckpt_name_C>
```

where:

*<rep\_session\_name\_AtoC>* = name of the replication session between A and C

*<ckpt\_name\_A>* = name of the user checkpoint for source VNX A

*<ckpt\_name\_C>* = name of the user checkpoint for destination VNX C

After the refresh completes, the user-created checkpoints for VNX A and VNX C have a common baseline for the replication pair. Essentially, now there is a common baseline between VNX A, VNX B, and VNX C.

Example:

```
[root@cse17107 nasadmin]# nas_replicate -refresh 17107s2-19105s2 -source otm_ckptA
-destination otm_ckptC
```

Output:

OK

4. To replace VNX A with VNX B in the replication topology, delete all replication sessions on A:
  - a. Delete the replication session between VNX A and VNX B by using the following command syntax on the source:

```
# nas_replicate -delete <rep_session_name_AtoB> -mode both
```

where:

*<rep\_session\_name\_AtoB>* = name of the replication session between VNX A and VNX B

Example:

```
[root@cse17107 nasadmin]# nas_replicate -delete 17107s2-17107s3 -mode both
```

Output:

OK

- b. Delete the replication session between VNX A and VNX C by using the following command syntax on the source:

```
# nas_replicate -delete <rep_session_name_AtoC> -mode both
```

where:

<rep\_session\_name\_AtoC> = name of the replication session between VNX A and VNX C

Example:

```
[root@cse17107 nasadmin]# nas_replicate -delete 17107s2-19105s2 -mode both
```

Output:

```
OK
```

5. Create a replication session from VNX B to VNX C by creating source and peer interconnects and then creating a replication session:

---

**Note:** You can also create interconnects by using Unisphere.

---

- a. Ensure that you have created a trust relationship between Control Station B and Control Station C by using the following command syntax for each VNX:

```
$ nas_cel -create <cel_name> -ip <ipaddr> -passphrase <passphrase>
```

where:

<cel\_name> = name of the remote VNX

<ipaddr> = IP address of the remote VNX's primary Control Station (in slot 0)

<passphrase> = passphrase used to manage the remote VNX

Example:

```
$ nas_cel -create vx17107 -ip 172.24.102.240 -passphrase nasdocs
```

Output:

```
operation in progress (not interruptible)...
id = 3
name = vx17107
owner = 0
device =
channel =
net_path = 172.24.102.240
VNX_id = APM000438070430000
passphrase = nasdocs
```

Run the -create command twice to ensure communication from both sides, first on the source VNX (to identify the destination VNX) and then on the destination VNX (to identify the source VNX).

---

**Note:** The passphrase should be the same on both the source and destination VNX.

---



- b. Create a Data Mover interconnect from B to C on the source VNX by using the following command syntax:

```
# nas_cel -interconnect -create <name_sourcetodest> -source_server
<MoverName_src> -destination_system <Destination_Control_Station_name>
-destination_server <MoverName_dst> -source_interface <interface IP_src>
-destination_interface <interface IP_dst>
```

where:

<name\_sourcetodest> = name of the interconnect for the source side

<MoverName\_src> = name of an available Data Mover on the local side of the interconnect

<Destination\_Control\_Station\_name> = name of the destination Control Station

<MoverName\_dst> = name of an available Data Mover on the peer side of the interconnect

<interface IP\_src>= IP address of an interface available on the local side of the interconnect

<interface IP\_dst>= IP address of an interface available on the peer side of the interconnect

Example:

```
# nas_cel -interconnect -create 17107s2-19105s2 -source_server server_2
-destination_system eng19105 -destination_server server_2 -source_interface
ip=10.245.17.110 -destination_interface ip=10.245.19.108
```

Output:

OK

- c. On the destination VNX, create the peer interconnect by using the following command syntax:

```
# nas_cel -interconnect -create <name_desttosource> -source_server
<MoverName_src> -destination_system <Source_Control_Station_name>
-destination_server <MoverName_dst> -source_interface <interface IP_src>
-destination_interface <interface IP_dst>
```

where:

<name\_desttosource> = name of the interconnect for the destination side

<MoverName\_src> = name of an available Data Mover on the local side of the interconnect

<Source\_Control\_Station\_name> = name of the source Control Station

<MoverName\_dst> = name of an available Data Mover on the peer side of the interconnect

<interface IP\_src>= IP address of an interface available on the local side of the interconnect

`<interface IP_dst>`= IP address of an interface available on the peer side of the interconnect

Example:

```
# nas_cel -interconnect -create 19105s2-17107s2 -source_server server_2
-destination_system cse17107 -destination_server server_2 -source_interface
ip=10.245.19.108 -destination_interface ip=10.245.17.110
```

Output:

OK

- d. On the source VNX, create the replication session from VNXB to VNXC by using the following command syntax:

```
# nas_replicate -create <rep_session_name_BtoC> -source -fs <src_fs_name>
-destination -fs <dst_fs_name> -interconnect <name_sourcetodest>
-max_time_out_of_sync <maxTimeOutOfSync>
```

where:

`<rep_session_name_BtoC>` = name of the replication session between B and C

`<src_fs_name>` = name of the source file system on VNX B

`<dst_fs_name>` = name of the destination file system on VNX C

`<name_sourcetodest>` = name of the interconnect for the source side from VNX B to VNX C

`<maxTimeOutOfSync>`= maximum time in minutes that the source and destination can be out of synchronization before an update occurs

The destination file system name should be the same as what was used in the previous replication session.

Example:

```
# nas_replicate -create 17107s3-19105s2 -source -fs otm_replica1 -destination
-fs otm -interconnect 17107s2-19105s2 -max_time_out_of_sync 10
```

Output:

OK

6. Immediately after the command prompt returns, verify that the initial synchronization is not performing a full copy by using the following command syntax on the source VNX:

```
$ nas_replicate -info <rep_session_name_BtoC>
```

where:

`<rep_session_name_BtoC>` = name of the replication session between VNX B and VNX C

Example:

```
[root@cse17107 nasadmin]# nas_replicate -info 17107s3-19105s2
```

Output:

```

ID =
245_FNM00103800639_2007_134_FNM00103600044_2007
Name = 17107s3-19105s2
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Wed Aug 24 13:23:37 EDT 2011
Type = filesystem
Celerra Network Server = eng19105
Dart Interconnect = 17107s2-19105s2
Peer Dart Interconnect = 19105s2-17107s2
Replication Role = source
Source Filesystem = otm_replica1
Source Data Mover = server_2
Source Interface = 10.245.17.110
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = otm
Destination Data Mover = server_2
Destination Interface = 10.245.19.108
Destination Control Port = 5085
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 10
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 0
Current Read Rate (KB/s) = 0
Current Write Rate (KB/s) = 0
Previous Transfer Rate (KB/s) = 93
Previous Read Rate (KB/s) = 57286
Previous Write Rate (KB/s) = 3858
Average Transfer Rate (KB/s) = 93
Average Read Rate (KB/s) = 57286
Average Write Rate (KB/s) = 3858

```

7. Verify that there is a current transfer taking place. The following fields should have values greater than zero:
  - Current Transfer Rate (KB/s)
  - Current Read Rate (KB/s)
  - Current Write Rate (KB/s)
  
8. Verify that the current transfer is not a full copy. The **Current Transfer is Full Copy** field should have the value **No**.

## Configure a cascading replication with common base checkpoints

1. Create a user checkpoint for each file system by using the following command syntax:

```
# fs_ckpt <fs_name> -name <ckpt_name> -Create
```

where:

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

`<ckpt_name>` = name of the user checkpoint created for the file system

---

**Note:** The `-name` command is optional. If you do not use the `-name` command to customize the names of the checkpoints, the system names the checkpoints. However, EMC recommends that you customize the names of the checkpoints for ease of management.

---

Example:

To create one user checkpoint each for three file systems:

```
[root@cse17107 nasadmin]# fs_ckpt cascade -name cascade_ckptA -Create
```

Output:

```

operation in progress (not interruptible)...id           = 107
name             = cascade
acl              = 0
in_use          = True
type            = uxfs
worm            = off
volume          = v245
pool            = clarata_archive
member_of       = root_avm_fs_group_10
rw_servers=     server_2
ro_servers=
rw_vdms         =
ro_vdms         =
auto_ext        = no,thin=no
deduplication   = Off
ckpts           =
root_rep_ckpt_107_53052_2,root_rep_ckpt_107_53052_1,cascade_ckptA
rep_sess        = 245_FNM00103800639_2007_254_FNM00103800639_2007(ckpts:
root_rep_ckpt_107_53052_1, root_rep_ckpt_107_53052_2)
stor_devs       = FNM00103800639-000F
disks           = d18
  disk=d18      stor_dev=FNM00103800639-000F addr=c16t1111
server=server_2
  disk=d18      stor_dev=FNM00103800639-000F addr=c0t1111
server=server_2
id              = 119
name            = cascade_ckptA
acl             = 0
in_use         = True
type           = ckpt
worm           = off
volume        = vp252
pool          = clarata_archive
member_of     =
rw_servers=
ro_servers=   server_2
rw_vdms       =
ro_vdms       =
ckpt_of=      cascade Wed Aug 24 11:11:52 EDT 2011
deduplication = Off
used           = 8%
full(mark)=   90%
stor_devs     = FNM00103800639-000F
disks         = d18
  disk=d18     stor_dev=FNM00103800639-000F addr=c16t1111
server=server_2
  disk=d18     stor_dev=FNM00103800639-000F addr=c0t1111
server=server_2

[root@cse17107 nasadmin]# fs_ckpt cascade_replica1 -name cascade_replica1_ckptB
-Create

```

Output:

```

operation in progress (not interruptible)...id           = 113
name           = cascade_replical
acl            = 0
in_use        = True
type          = uxfs
worm          = off
volume        = v254
pool          = clarata_archive
member_of     = root_avm_fs_group_10
rw_servers=
ro_servers= server_3
rw_vdms       =
ro_vdms       =
auto_ext      = no,thin=no
deduplication = Off
ckpts         = root_rep_ckpt_113_53065_2,root_rep_ckpt_113_53065_1,
               root_rep_ckpt_113_53280_2,root_rep_ckpt_113_53280_1,
               cascade_replical_ckptB
rep_sess      = 245_FNM00103800639_2007_254_FNM00103800639_2007(ckpts:
root_rep_ckpt_113_53065_1, root_rep_ckpt_113_53065_2),
254_FNM00103800639_2007_129_FNM00103600044_2007(ckpts:
root_rep_ckpt_113_53280_1, root_rep_ckpt_113_53280_2)
stor_devs     = FNM00103800639-0009
disks         = d12
  disk=d12    stor_dev=FNM00103800639-0009 addr=c16t115
server=server_3
  disk=d12    stor_dev=FNM00103800639-0009 addr=c0t115
server=server_3
id            = 120
name          = cascade_replical_ckptB
acl           = 0
in_use        = True
type          = ckpt
worm          = off
volume        = vp257
pool          = clarata_archive
member_of     =
rw_servers=
ro_servers= server_3
rw_vdms       =
ro_vdms       =
ckpt_of= cascade_replical Wed Aug 24 11:13:02 EDT 2011
deduplication = Off
used          = 13%
full(mark)= 90%
stor_devs     = FNM00103800639-000D
disks         = d16
  disk=d16    stor_dev=FNM00103800639-000D addr=c16t119
server=server_3
  disk=d16    stor_dev=FNM00103800639-000D addr=c0t119
server=server_3

```

---

**Note:** The following checkpoint creation is on the destination side. The name of the file system is the same as the source because of replication.

---

```

[nasadmin@ENG19105 ~]$ fs_ckpt cascade_replical -name cascade_replical_ckptC
-Create

```

Output:

```

operation in progress (not interruptible)...id          = 113
name           = cascade_replica1
acl            = 0
in_use        = True
type          = uxfs
worm          = off
volume        = v129
pool          = clarsas_archive
member_of     = root_avm_fs_group_32
rw_servers=
ro_servers= server_2
rw_vdms       =
ro_vdms       =
auto_ext      = no,thin=no
deduplication = Off
ckpts         =
root_rep_ckpt_113_9605_2,root_rep_ckpt_113_9605_1,cascade_replica1_ckptC
rep_sess      = 254_FNM00103800639_2007_129_FNM00103600044_2007(ckpts:
root_rep_ckpt_113_9605_1, root_rep_ckpt_113_9605_2)
stor_devs     = FNM00103600044-0000
disks         = d8
disk=d8       stor_dev=FNM00103600044-0000 addr=c16t110
server=server_2
disk=d8       stor_dev=FNM00103600044-0000 addr=c0t110
server=server_2
id            = 117
name          = cascade_replica1_ckptC
acl           = 0
in_use        = True
type          = ckpt
worm          = off
volume        = vp132
pool          = clarsas_archive
member_of     =
rw_servers=
ro_servers= server_2
rw_vdms       =
ro_vdms       =
ckpt_of= cascade_replica1 Wed Aug 24 11:13:55 EDT 2011
deduplication = Off
used          = 8%
full(mark)= 90%
stor_devs     = FNM00103600044-0000
disks         = d8
disk=d8       stor_dev=FNM00103600044-0000 addr=c16t110
server=server_2
disk=d8       stor_dev=FNM00103600044-0000 addr=c0t110
server=server_2

```

2. On the source VNX, refresh the local replication session from VNX A to VNX B with the user-created checkpoints by using the following command syntax:

```
# nas_replicate -refresh <rep_session_name_AtoB> -source <ckpt_name_A>
-destination <ckpt_name_B>
```

where:

<rep\_session\_name\_AtoB> = name of the replication session between VNX A and VNX B

<ckpt\_name\_A> = name of the user checkpoint for source VNX A

<ckpt\_name\_B> = name of the user checkpoint for destination VNX B

After the refresh completes, the user-created checkpoints for VNX A and VNX B have a common baseline for the replication pair.

Example:

```
[root@cse17107 nasadmin]# nas_replicate -refresh S2_107-S3_107 -source
cascade_ckptA -destination cascade_replica1_ckptB
```

Output:

OK

3. On the source VNX, refresh the remote replication session from VNX B to VNX C with the user-created checkpoints by using the following command syntax:

```
# nas_replicate -refresh <rep_session_name_BtoC> -source <ckpt_name_B>
-destination <ckpt_name_C>
```

where:

<rep\_session\_name\_BtoC> = name of the replication session between VNX B and VNX C

<ckpt\_name\_B> = name of the user checkpoint for source VNX B

<ckpt\_name\_C> = name of the user checkpoint for destination VNX C

After the refresh completes, the user-created checkpoints for VNX B and VNX C have a common baseline for replication pair. Essentially, now there is a common baseline between VNX A, VNX B, and VNX C.

Example:

```
#nas_replicate -refresh S3_141-S219105 -source cascade_replica1_ckptB -destination
cascade_replica1_ckptC
```

Output:

OK

4. To replace VNX B with VNX C in the replication topology, delete all replication sessions on VNX B:
  - a. Delete the replication session between A and B by using the following command syntax on the source:

```
# nas_replicate -delete <rep_session_name_AtoB> -mode both
```

where:

<rep\_session\_name\_AtoB> = name of the replication session between A and B

Example:

```
# nas_replicate -delete S2_107-S3_107 -mode both
```

Output:

OK



- b. Delete the replication session between B and C by using the following command syntax on the source:

```
# nas_replicate -delete <rep_session_name_BtoC> -mode both
```

where:

<rep\_session\_name\_BtoC> = name of the replication session between VNX B and VNX C

Example:

```
# nas_replicate -delete S3_141-S219105 -mode both
```

Output:

```
OK
```

5. Create a replication session from VNX A to VNX C by creating source and peer interconnects and then creating a replication session:

---

**Note:** You can also create interconnects by using Unisphere.

---

- a. Ensure that you have created a trust relationship between Control Station A and Control Station C by using the following command syntax for each VNX:

```
$ nas_cel -create <cel_name> -ip <ipaddr> -passphrase <passphrase>
```

where:

<cel\_name> = name of the remote VNX

<ipaddr> = IP address of the remote VNX's primary Control Station (in slot 0)

<passphrase> = passphrase used to manage the remote VNX

Example:

```
$ nas_cel -create vx17107 -ip 172.24.102.240 -passphrase nasdocs
```

Output:

```
operation in progress (not interruptible)...
id = 3
name = vx17107
owner = 0
device =
channel =
net_path = 172.24.102.240
VNX_id = APM000438070430000
passphrase = nasdocs
```

Run the -create command twice to ensure communication from both sides, first on the source VNX (to identify the destination VNX) and then on the destination VNX (to identify the source VNX).

---

**Note:** The passphrase should be the same on both the source and destination VNX.

---

- b. Create a Data Mover interconnect from A to C on the source VNX by using the following command syntax:

```
# nas_cel -interconnect -create <name_sourcetodest> -source_server
<MoverName_src> -destination_system <Destination_Control_Station_name>
-destination_server <MoverName_dst> -source_interface <interface IP_src>
-destination_interface <interface IP_dst>
```

where:

<name\_sourcetodest> = name of the interconnect for the source side

<MoverName\_src> = name of an available Data Mover on the local side of the interconnect

<Destination\_Control\_Station\_name> = name of the destination Control Station

<MoverName\_dst> = name of an available Data Mover on the peer side of the interconnect

<interface IP\_src>= IP address of an interface available on the local side of the interconnect

<interface IP\_dst>= IP address of an interface available on the peer side of the interconnect

Example:

```
# nas_cel -interconnect -create 17107s2-19105s2 -source_server server_2
-destination_system eng19105 -destination_server server_2 -source_interface
ip=10.245.17.110 -destination_interface ip=10.245.19.108
```

Output:

OK

- c. On the destination VNX, create the peer interconnect by using the following command syntax:

```
# nas_cel -interconnect -create <name_desttosource> -source_server
<MoverName_src> -destination_system <Source_Control_Station_name>
-destination_server <MoverName_dst> -source_interface <interface IP_src>
-destination_interface <interface IP_dst>
```

where:

<name\_desttosource> = name of the interconnect for the destination side

<MoverName\_src> = name of an available Data Mover on the local side of the interconnect

<Source\_Control\_Station\_name> = name of the source Control Station

<MoverName\_dst> = name of an available Data Mover on the peer side of the interconnect

<interface IP\_src>= IP address of an interface available on the local side of the interconnect

`<interface IP_dst>`= IP address of an interface available on the peer side of the interconnect

Example:

```
# nas_cel -interconnect -create 19105s2-17107s2 -source_server server_2
-destination_system cse17107 -destination_server server_2 -source_interface
ip=10.245.19.108 -destination_interface ip=10.245.17.110
```

Output:

OK

- d. On the source VNX, create the replication session from A to C by using the following command syntax:

```
# nas_replicate -create <rep_session_name_AtoC> -source -fs <src_fs_name>
-destination -fs <dst_fs_name> -interconnect <name_sourcetodest>
-max_time_out_of_sync <maxTimeOutOfSync>
```

where:

`<rep_session_name_AtoC>` = name of the replication session between A and C

`<src_fs_name>` = name of the source file system on A

`<dst_fs_name>` = name of the destination file system on C

`<name_sourcetodest>` = name of the interconnect for the source side from A to C

`<maxTimeOutOfSync>`= maximum time in minutes that the source and destination can be out of synchronization before an update occurs

The destination file system name should be the same as what was used in the previous replication session.

Example:

```
[root@cse17107 nasadmin]# nas_replicate -create cascade_17107s2-19105s2 -source
-fs cascade -destination -fs cascade_replica1 -interconnect 17107s2-19105s2
-max_time_out_of_sync 10
```

Output:

OK

6. Immediately after the command prompt returns, verify that the initial synchronization is not performing a full copy by using the following command syntax on the source VNX:

```
$ nas_replicate -info <rep_session_name_AtoC>
```

where:

`<rep_session_name_AtoC>` = name of the replication session between A and C

Example:

```
[root@cse17107 nasadmin]# nas_replicate -i cascade_17107s2-19105s2
```

Output:

```

ID =
245_FNM00103800639_2007_129_FNM00103600044_2007
Name = cascade_17107s2-19105s2
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Wed Aug 24 12:31:11 EDT 2011
Type = filesystem
Celerra Network Server = eng19105
Dart Interconnect = 17107s2-19105s2
Peer Dart Interconnect = 19105s2-17107s2
Replication Role = source
Source Filesystem = cascade
Source Data Mover = server_2
Source Interface = 10.245.17.110
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = cascade_replica1
Destination Data Mover = server_2
Destination Interface = 10.245.19.108
Destination Control Port = 5085
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 10
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 0
Current Read Rate (KB/s) = 0
Current Write Rate (KB/s) = 0
Previous Transfer Rate (KB/s) = 42
Previous Read Rate (KB/s) = 54613
Previous Write Rate (KB/s) = 3831
Average Transfer Rate (KB/s) = 42
Average Read Rate (KB/s) = 54613
Average Write Rate (KB/s) = 3831

```

7. Verify that there is a current transfer taking place. The following fields should have values greater than zero:
  - Current Transfer Rate (KB/s)
  - Current Read Rate (KB/s)
  - Current Write Rate (KB/s)
  
8. Verify that the current transfer is not a full copy. The **Current Transfer is Full Copy** field should have the value **No**.

## Configure the replication of a file system mounted on an Nested Mount File System mountpoint

A Nested Mount File System (NMFS) allows you to manage a collection of component file systems as a single file system. CIFS and NFS clients see component file systems as a single share or single export.

To replicate a file system mounted on an NMFS mount point:

1. Create a replication session from source to destination.
2. Stop the replication session of the file system either by using the `-stop` mode option on both the source and destination simultaneously, or by using the `-stop` mode option on the source and destination separately.
3. Create the NMFS at the Disaster Recovery site.
4. Unmount the destination file system.
5. Mount the NMFS.
6. Remount the destination file system under the NMFS path as read only.
7. Start the replication session from the source while selecting the **Discard the changes since last copy** in Unisphere, or by using the `-overwrite_destination` option from the command line.



## Managing replication sessions

After you have created a replication session, you can manage the session.

The online VNX for file man pages and the *EMC VNX Command Line Interface Reference for File* provide a detailed synopsis of the commands and syntax conventions presented in this section.

The tasks to manage replication sessions are:

- ◆ [Get information about replication sessions on page 128](#)
- ◆ [Modify replication properties on page 131](#)
- ◆ [Refresh the destination on page 132](#)
- ◆ [Delete a replication session on page 133](#)
- ◆ [Stop a replication session on page 134](#)
- ◆ [Start a replication session on page 136](#)
- ◆ [Start a failed over or switched over replication on page 138](#)
- ◆ [Start a replication session that is involved in a one-to-many configuration on page 139](#)
- ◆ [Reverse the direction of a replication session on page 140](#)
- ◆ [Switch over a replication session on page 141](#)
- ◆ [Fail over a replication session on page 144](#)

## Get information about replication sessions

You can view a list of all configured or stopped replication sessions on a Data Mover in the VNX for file cabinet and obtain detailed status information for a specific session or for all sessions.

Action					
To view a list of replication sessions, use this command syntax:					
<code>\$ nas_replicate -list</code>					
Output					
Name	type	Local Mover	Interconnect	Celerra	Status
fs1_12_13	filesystem	server_2	-->20003	eng25213	OK
fs2_41	filesystem	server_3	<--40s3-41s2	eng25241	OK
vdm2-to-41	vdm	server_2	-->40s2-41s2	eng25241	OK
rep5_cpy	copy	server_2	-->40s2-41s2	eng25241	OK
f3-cpy-41	copy	server_2	-->40s2-41s2	eng25241	OK

To view detailed status information on a specific replication session, use the `-info` command.

Action
To display detailed information on a specific replication session, use this command syntax:
<code>\$ nas_replicate -info {id=&lt;sessionId&gt; &lt;name&gt;}</code>
where:
<code>&lt;sessionId&gt;</code> = ID of the replication session
<code>&lt;name&gt;</code> = name of the replication session
<b>Note:</b> To obtain the session ID for the first time, run <code>nas_replicate -info</code> and specify the replication name.
Example:
To display information about file system replication session <code>fs1_12_13</code> , type:
<code>\$ nas_replicate -info fs1_12_13</code>



**Output**

**Note:** This output shows a file system replication.

```

ID =
175_APM00061205365_0000_90_APM00064805756_0000
Name = fs1_12_13
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Mon Oct 01 17:38:58 EDT 2007
Type = filesystem
Celerra Network Server = eng25213
Dart Interconnect = 20003
Peer Dart Interconnect = 20003
Replication Role = source
Source Filesystem = fs1
Source Data Mover = server_2
Source Interface = 172.24.252.26
Source Control Port = 0
Destination Filesystem = fs2
Destination Data Mover = server_2
Destination Interface = 172.24.252.27
Destination Control Port = 5081
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 20
Next Transfer Size (KB) = 0
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 1
Current Read Rate (KB/s) = 32251
Current Write Rate (KB/s) = 303
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 3
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

```

The system displays the appropriate output depending on the type of source object replicated.

**Output definitions**

The definitions are:

ID — The fixed ID assigned to this replication session.

Name — Name assigned to this replication session.

Source Status — Indicates the source status of a replication relationship.

Network Status — Displays the network status for this replication session.

Destination Status — Indicates the destination status of a replication relationship.

Last Sync Time (GMT) — Indicates when data was copied to the destination file system. Time displays in GMT.

Type — Replication type. Displays file system, vdm, or copy.

Celerra Network Server — Displays the Control Station name of the destination system used in this replication relationship.

Dart Interconnect — Name of the interconnect used by the source in this replication session.

Peer Dart Interconnect — Name of the interconnect used by the other (peer) side in this replication.

Replication Role — Indicates whether the role is as source or destination.

Source File System — Source file system name.

Source Data Mover — Source Data Mover in the replication session.

Source Interface — Source site interface (IP address) used to transport this replication session data.

Source Control Port — Identifies the source Control Station port.

Source Current Data Port — Identifies the source data port used for the current transfer.

Destination File System — Destination file system name.

Destination Data Mover — Destination Data Mover in the replication session.

Destination Interface — Destination site interface (IP address) used to transport this replication session data.

Destination Control Port — Identifies the destination Control Station port.

Destination Data Port — Identifies the destination port receiving the data.

Source VDM — For VDM replication, source VDM for this replication session.

Max Out of Sync Time (minutes) — If the update policy is `max_time_out_of_sync`, then this field defines the time, in minutes, before an update occurs.

Next Transfer Size (KB) — Identifies the size of the next transfer in kilobytes.

Current Transfer Size (KB) — Identifies the total size of the data to transfer in kilobytes.

Current Transfer Remain (KB) — Identifies how much data, in kilobytes, is left in the current transfer.

Estimated Completion Time — Estimates a time at which the current transfer completes.

**Output definitions**

Current Transfer is a Full Copy — Indicates whether the current transfer is a full copy instead of a differential copy.

Current Transfer Rate (KB/s) — Identifies the current transfer rate in kilobytes per second.

Current Read Rate (KB/s) — Identifies the current read rate in kilobytes per second.

Current Write Rate (KB/s) — Identifies the current write rate in kilobytes per second.

Previous Transfer Rate (KB/s) — Identifies the rate of the previous transfer in kilobytes per second.

Previous Read Rate (KB/s) — Identifies the previous read rate in kilobytes per second.

Previous Write Rate (KB/s) — Identifies the previous write rate in kilobytes per second.

Average Transfer Rate (KB/s) — Identifies the average rate of transfer in kilobytes per second.

Average Read Rate (KB/s) — Identifies the average read rate in kilobytes per second.

Average Write Rate (KB/s) — Indicates the average write rate in kilobytes per second.

## Modify replication properties

You can change the name of a replication session, the source and destination interfaces used, and the `max_time_out_of_sync` value.

**Action**

To change the name of a replication session, use this command syntax:

```
$ nas_replicate -modify {<name>|id=<sessionId>} -name <newName>
```

where:

<name> = name of the replication session to modify

<sessionId> = ID of the replication session to modify

<newName> = new name of the replication session

Example:

To change the name of replication session `fs1_12_13` to `Repl_from_12_13`, type:

```
$ nas_replicate -modify fs1_12_13 -name Repl_from_12_to_13
```

**Output**

OK

**Verification**

To verify that the replication name was changed, type:

```
$ nas_replicate -list
```

Output:

Name	type	Local Mover	Interconnect	Celerra	Status
Repl_from_12_to_13	filesystem	server_2	-->20003	eng25213	OK

**Verification**

To verify from the destination side, type:

```
$ nas_replicate -list
```

Output:

Name	type	Local Mover	Interconnect	Celerra	Status
Repl_from_12_to_13	filesystem	server_2	<--20003	eng25212	OK

**Refresh the destination**

From the source side of a replication session, you can perform a manual refresh of a destination object to capture updates made on a source object. Refreshing a replication session updates the destination side of the specified replication session based on changes to the source side. This has the same effect as reaching the `max_time_out_of_sync` value.

To update the destination object automatically, use the `max_time_out_of_sync` value. [Updating the destination site with source changes on page 43](#) explains this option in detail.

**Action**

To update the destination side of a specific replication session manually, use this command syntax:

```
$ nas_replicate -refresh {<name>|id=<sessionid>} [-background]
```

where:

`<name>` = name of the replication session

`<sessionid>` = ID of the replication session

Example:

To refresh the destination side of session `fs1_12_13`, type:

```
$ nas_replicate -refresh fs1_12_13 -background
```

When data changes on the source are large, the `-refresh` command can take a long time to complete. This example shows running this command in background mode.

**Note:** Execute this command from the Control Station on the source side only.

**Output**

```
Info [26843676673]: In Progress: Operation is still
running. Check task id 2663 on the Task Status screen for results.
```

**Verification**

To verify the progress of the refresh, use the `nas_task` command and type:

```
$ nas_task -info 2663
```

Verification	
Output:	
Task Id	= 2663
Celerra Network Server	= eng25212
State	= Succeeded
Description	= Refresh Replication
NONE:id=181_APM00061205365_0000_108_APM00064805756_0000.	
Originator	= nasadmin@cli.localhost
Start Time	= Mon Oct 01 18:42:04 EDT 2007
End Time	= Mon Oct 01 18:42:09 EDT 2007
Schedule	= n/a
Response Statuses	= OK

## Delete a replication session

You would delete a session when you no longer want to replicate the file system or VDM.

If you delete a replication session by using the `-both` or `-destination` mode options, and data transfer is in progress, the system will perform an internal checkpoint restore operation of the latest checkpoint on the destination side. This will undo any changes written to the file system as part of the current data transfer and bring the file system back to a consistent state. If the checkpoint restore fails, for example there was not enough SavVol space, the destination file system will remain in an inconsistent state and client-based access should be avoided.

---

**Note:** Deleting a replication does not delete the underlying source objects. However, internal checkpoints that are part of the session are deleted.

---

Action
To delete both sides of a remote replication session from the source side, use this command syntax:
<pre>\$ nas_replicate -delete {&lt;name&gt; id=&lt;sessionId&gt;} -mode {both}</pre>
where:
<code>&lt;name&gt;</code> = name of the replication session to delete
<code>&lt;sessionId&gt;</code> = ID of the replication session to delete
Example:
To delete both sides of replication session fs1_12_13, type:
<pre>\$ nas_replicate -delete fs1_12_13 -mode both</pre>
Output
OK

When communication is down, use the source and destination mode options.

**Action**

To delete the replication session from the source side only, use this command syntax:

```
$ nas_replicate -delete {<name>|id=<sessionId>} -mode {source}
```

where:

<name> = name of the replication session to delete

<sessionId> = ID of the replication session to delete

Example:

To delete the source side only of replication session fs1\_12\_13, type:

```
$ nas_replicate -delete fs1_12_13 -mode source
```

**Output**

OK

The destination mode option deletes the replication session from the destination side only.

**Action**

To delete the replication session from the destination side only, use this command syntax:

```
$ nas_replicate -delete {<name>|id=<sessionId>} -mode {destination}
```

where:

<name> = name of the replication session to delete

<sessionId> = ID of the replication session to delete

Example:

To delete the destination side only of replication session fs1\_12\_13, from the destination side, type:

```
$ nas_replicate -delete fs1_12_13 -mode destination
```

**Output**

OK

## Stop a replication session

You cannot stop a session if -delete is already running for the session. After a stop operation is in progress, only -list and -info are permitted, as well as any nas\_task command. [Stopping a replication session on page 44](#) describes how the stop operation works.

**Note:** While a session is stopped, data continues to accumulate in the SavVol. If you do not plan to start the session again, consider deleting the session.

If you stop a replication session by using the -both or -destination mode options, and data transfer is in progress, the system will perform an internal checkpoint restore operation of the latest checkpoint on the destination side. This will undo any changes written to the file

system as part of the current data transfer and bring the file system back to a consistent state. If the checkpoint restore fails, for example there was not enough SavVol space, the destination file system will remain in an inconsistent state and client-based access should be avoided.

Action
<p>To stop replication on both source and destination file systems simultaneously, use this command syntax from the source site:</p> <pre>\$ nas_replicate -stop {&lt;name&gt; id=&lt;session_id&gt;} -mode {both}</pre> <p>where:</p> <p>&lt;name&gt; = name of the replication session to stop</p> <p>&lt;session_id&gt; = ID of the replication session to stop</p> <p>Example:</p> <p>To stop replication fs1_12_13, type:</p> <pre>\$ nas_replicate -stop fs1_12_13 -mode both</pre>
Output
OK

Verification
<p>To verify that replication session fs1_12_13 is stopped, on the source side, type:</p> <pre>\$ nas_replicate -list</pre> <p>Output:</p> <pre>Name          type          Local Mover  Interconnect Celerra  Status fs1_12_13     filesystem    server_2    --&gt;20003      eng25212  Info 26045317429: Replication session is stopped. Stopped replication session keeps replication configuration information and internal snaps.</pre> <p>To verify that the session is stopped on the destination side, type:</p> <pre>\$ nas_replicate -list</pre> <p>Output:</p> <pre>Name          type          Local Mover  Interconnect Celerra  Status fs1_12_13     filesystem    server_2     &lt;--20003     eng25212  OK</pre>

If communication is down, use the source and destination options to stop the session.

Action
<p>To stop replication on either the source or destination file systems, use this command syntax from the source site:</p> <pre>\$ nas_replicate -stop {&lt;name&gt; id=&lt;session_id&gt;} -mode {source destination}</pre> <p>where:</p>

Action
<p><code>&lt;name&gt;</code> = name of the replication session to stop</p> <p><code>&lt;session_id&gt;</code> = ID of the replication session to stop</p> <p>Example:</p> <p>To stop replication fs1_12_13 on the source side only, type:</p> <pre>\$ nas_replicate -stop fs1_12_13 -mode source</pre> <p>To stop replication fs1_12_13 on the destination side only, type:</p> <pre>\$ nas_replicate -stop fs1_12_13 -mode destination</pre>
Output
OK

Verification
<p>To verify that replication session fs1_12_13 is stopped on the source side, type:</p> <pre>\$ nas_replicate -list</pre> <p>Output:</p> <pre>Name      type      Local Mover  Interconnect Celerra      Status fs1_12_13 filesystem server_2     --&gt;20003     eng25212     Info 26045317429: Replication session is stopped. Stopped replication session keeps replication configuration information and internal snaps.</pre> <p>To verify that the session is stopped on the destination side, from the destination side, type:</p> <pre>\$ nas_replicate -list</pre> <p>Output:</p> <pre>Name      type      Local Mover  Interconnect Celerra      Status fs1_12_13 filesystem server_2     &lt;--20003     eng25212     OK</pre>

## Start a replication session

The `-start` option allows you to start a replication session again by using an incremental or a full data copy. Use the `-start` option to start a stopped, failed over, or switched over replication session.

After you stop a replication session, only the `-start` option can start it again. This command verifies whether replication is in a condition that allows it to start again. If you are using this procedure to change interconnect or interfaces, or to change the update policy, specify them when you start the replication relationship.

Action
To start a file system replication session, use this command syntax from the source site:



Action
<pre>\$ nas_replicate -start {&lt;name&gt; id=&lt;sessionId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the replication session</p> <p>&lt;session_id&gt; = session ID. This is required if a duplicate replication session is detected</p> <p>Example:</p> <p>To start replication fs1_12_13, type:</p> <pre>\$ nas_replicate -start fs1_12_13</pre>
Output
OK

Verification																								
<p>To verify that the session was started on the source side, type:</p> <pre>\$ nas_replicate -list</pre> <table border="1"> <thead> <tr> <th>Name</th> <th>type</th> <th>Local M</th> <th>Interconnect</th> <th>Celerra</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>fs1_12_13</td> <td>filesystem</td> <td>server_2</td> <td>--&gt;20003</td> <td>eng25213</td> <td>OK</td> </tr> </tbody> </table> <p>To verify that the session was started on the destination side, from the destination side, type:</p> <pre>\$ nas_replicate -list</pre> <p>Output:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>type</th> <th>Local M</th> <th>Interconnect</th> <th>Celerra</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>fs1_12_13</td> <td>filesystem</td> <td>server_2</td> <td>--&gt;20003</td> <td>eng25212</td> <td>OK</td> </tr> </tbody> </table>	Name	type	Local M	Interconnect	Celerra	Status	fs1_12_13	filesystem	server_2	-->20003	eng25213	OK	Name	type	Local M	Interconnect	Celerra	Status	fs1_12_13	filesystem	server_2	-->20003	eng25212	OK
Name	type	Local M	Interconnect	Celerra	Status																			
fs1_12_13	filesystem	server_2	-->20003	eng25213	OK																			
Name	type	Local M	Interconnect	Celerra	Status																			
fs1_12_13	filesystem	server_2	-->20003	eng25212	OK																			

This example changes the interconnect used for the replication session.

Action
<p>To start a file system replication session and change the interconnect, use this command syntax from the source site:</p> <pre>\$ nas_replicate -start {&lt;name&gt; id=&lt;session_id&gt;} -interconnect {&lt;name&gt; id=&lt;interConnectId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = replication session name</p> <p>&lt;session_id&gt; = session ID. This is required if a duplicate replication session is detected</p> <p>&lt;name&gt; = name of the established source-side Data Mover interconnect to use for the replication session</p> <p>&lt;interConnectID&gt; = ID of the interconnect to use</p> <p>Example:</p> <p>To start replication session fs1_12_13 and change the interconnect, type:</p> <pre>\$ nas_replicate -start fs1_12_13 -interconnect cs110_s3s2</pre>

**Output**

OK

This example changes the update policy for the session.

**Action**

To start a file system replication session and change the default time of out of sync value, use this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>} -max_time_out_of_sync
<max_time_out_of_sync>
```

where:

<name> = replication session name

<session\_id> = session ID. This is required if a duplicate replication session is detected

<max\_time\_out\_of\_sync> = time, in 1–1440 minutes (up to 24 hours), that the source and destination can be out of synchronization before an update occurs

Example:

To start replication session fs1\_12\_13 and set a max\_time\_out\_of\_sync of 10 minutes, type:

```
$ nas_replicate -start fs1_12_131 -max_time_out_of_sync 10
```

**Output**

OK

## Start a failed over or switched over replication

Use the `nas_replicate -start` with `-reverse` option to start a replication session that has failed over or switched over.

### Before starting a failed over replication that uses VDMs

If you are starting a failed over replication that uses VDMs in a CIFS environment, verify that the source site is available. When you are ready to start the session, do so in the following order:

1. Start the failed over VDM replication session by using the `-reverse` option.
2. Start the replication session of the file system contained in the VDM by using the `-reverse` option.
3. Reverse the direction of the VDM replication session. When the reverse completes, the VDM state on the source site changes from unloaded to loaded and on the destination site from loaded to mounted.
4. Reverse the file system replication session contained in the VDM.

**Action**

To start a replication session after failover or switchover, use this command syntax:

Action
<pre>\$ nas_replicate -start {&lt;name&gt; id=&lt;session_id&gt;} -reverse</pre> <p>where:</p> <p>&lt;name&gt; = replication session name</p> <p>&lt;session_id&gt; = session ID. This is required if a duplicate replication session is detected</p> <p>Example:</p> <p>To start replication session fs1_12_13 and reverse the direction of the session, type:</p> <pre>\$ nas_replicate -start fs1_12_13 -reverse -overwrite_destination</pre>
Output
OK

## Start a replication session that is involved in a one-to-many configuration

This section describes how to start switched over or failed over sessions when the source object is used in more than one replication session.

### Start a switched over session

To start a switched over replication session:

1. Select the replication session from which you want to resynchronize.
2. Start the replication session identified in step 1 from the original destination side and choose the `-reverse` option. [Start a failed over or switched over replication on page 138](#) provides details on how to start the session.
3. Reverse that replication session. [Reverse the direction of a replication session on page 140](#) provides details on how to reverse the session.
4. If you switched over multiple sessions at one time, change the mount status of the destination objects in the remaining sessions. [Change the mount status of source or destination object on page 177](#) provides details on how to change the mount status of the destination objects.
5. Start the remaining sessions from the original source side. [Start a replication session on page 136](#) provides details on how to start each session. When starting each session, ensure you specify the `-overwrite_destination` option. Do not specify the `-reverse` option.

### Start a failed over session

You can keep changes from only one failed-over replication session per source object. If you fail over multiple sessions for a given object, you can only save the changes from one of the failed-over replication sessions.

---

**Note:** Before you start, ensure that the source site is now available.

---

To start the failed over replication session:

1. Select the replication session from which you want to resynchronize.
2. Start the replication session identified in step 1 from the original destination side and specify the `-reverse` and `overwrite_destination` options. [Start a failed over or switched over replication on page 138](#) provides details on how to start the session.
3. Reverse that replication session. [Reverse the direction of a replication session on page 140](#) provides details on how to reverse the session.
4. If you failed over multiple sessions, change the mount status of the destination objects to read-only in the remaining sessions. [Change the mount status of source or destination object on page 177](#) provides details on how to change the mount status of objects.
5. Start the remaining sessions from the original source side. [Start a replication session on page 136](#) provides details on how to start each session. When starting each session, ensure you specify the `-overwrite_destination` option. Do not choose the `-reverse` option.

## Reverse the direction of a replication session

Use a replication reversal to change the direction of replication. You may want to change the direction of replication to perform maintenance on the source site or to do testing on the destination site.

You can only reverse a replication from the source side.

If you reverse a replication session that is involved in a one-to-many configuration, the source side goes into cascade mode. The destination side from one of the one-to-many sessions becomes the source and the original source side becomes the destination and that source cascades out to other destination sessions.

Action
<p>To change the direction of a remote replication session, use this command syntax:</p> <pre>\$ nas_replicate -reverse {&lt;name&gt; id=&lt;sessionId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the replication session to reverse</p> <p>&lt;sessionId&gt; = ID of the replication session to reverse</p> <p>Example:</p> <p>For the current read/write file system, fs1_12_13, to become the read-only file system, type:</p> <pre>\$ nas_replicate -reverse fs1_12_13</pre>
Output
OK

**Notes**

When this command completes, the current read/write file system (fs1\_12\_13) becomes read-only and the current read-only file system (dst\_ufs1) becomes read/write.

If you tried to run this command from the incorrect side (read-only), the following error message appears:

```
Error 2247: this command must be issued on the current
source site:cs100
```

**Verification**

To verify that the direction of the replication reversed, type:

```
$ nas_replicate -list
```

Before the reverse operation, type from the source side:

```
$ nas_replicate -list
```

Output:

Name	type	Local	Mover	Interconnect	Celerra	Status
fs1_12_13	filesystem		server_2	-->	20003	eng25213 OK

From the destination side, type:

```
$ nas_replicate -list
```

Output:

Name	type	Local	Mover	Interconnect	Celerra	Status
fs1_12_13	filesystem		server_2	<--	20003	eng25212 OK

After the reverse operation, from the source side type:

```
$ nas_replicate -list
```

Output:

Name	type	Local	Mover	Interconnect	Celerra	Status
fs1_12_13	filesystem		server_2	<--	20003	eng25213 OK

From the destination side, type:

```
$ nas_replicate -list
```

Output:

Name	type	Local	Mover	Interconnect	Celerra	Status
fs1_12_13	filesystem		server_2	-->	20003	eng25212 OK

## Switch over a replication session

Perform a replication switchover from the source side only when both the source and destination sides are available.

The switchover operation stops a replication session after synchronizing the destination object with the source object without data loss. The operation mounts the source object as

read-only, and marks the destination object as read/write so that it can act as the new source object.

#### Before you execute this command

Note the following:

- ◆ Ensure that clients have access to the destination Data Mover so clients can access the destination object immediately.
  - ◆ Switchover does not change the defined role of the source and destination objects, only the mount status of the object (read-only and read/write) in the replication.
  - ◆ When the switchover completes, if you want to start the replication session again, you must use the `-start` with `-reverse` option so that the defined role of the object matches the new mount status.
  - ◆ To return to the original replication configuration and start the session in the proper direction, perform a `nas_replicate -reverse`.
1. Before executing the switchover, verify replication session information on the source and destination, by using this command syntax:

```
$ nas_replicate -list
```

Source output:

Name	type	Local Mover	Interconnect	Celerra	Status
fs1_1213	filesystem	server_2	-->20003	eng25213	OK

Destination output:

Name	type	Local Mover	Interconnect	Celerra	Status
fs1_1213	filesystem	server_2	<--20003	eng25212	OK

2. Perform a switchover by using this command syntax:

```
$ nas_replicate -switchover {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session to switchover

`<session_id>` = ID of the replication session to switchover

Example:

To perform a switchover of replication session `fs1_1213`, from the source, type:

```
$ nas_replicate -switchover fs1_1213
```

Output:

```
OK
```

---

**Note:** Execute this command from the Control Station on the source VNX only.

---

3. Verify the switchover, from the source side and destination side by using this command syntax:

```
$ nas_replicate -list
```

Source output:

```
Name type Local Mover Interconnect Celerra Status
fs1_1213 filesystem server_2 -->20003 eng25212 Info
26045317429: Replication session is stopped.
Stopped replication session keeps replication
configuration information and internal snaps.
```

Destination output:

```
Name type Local Mover Interconnect Celerra Status
fs1_1213 filesystem server_2 <--20003 eng25212 Info
26045317429: Replication session is stopped.
Stopped replication session keeps replication
configuration information and internal snaps.
```

4. Optionally, verify the new mount status of the source object and destination object by using this command syntax:

```
$ server_mount <movername>
```

where:

<movername> = the Data Mover where the file system is mounted

Example:

To display the mounted file systems on Data Mover server\_2 and verify that the source file system fs1 is mounted read-only and the destination file system fs2 is mounted read/write, from the source side, type:

```
$ server_mount server_2
```

Source output:

```
server_2 :
root_fs_2 on / uxfs,perm,rw
root_fs_common on /.etc_common uxfs,perm,ro
fs2 on /fs2 uxfs,perm,ro
fs2_ckpt1 on /fs2_ckpt1 ckpt,perm,ro
fs1 on /fs1 uxfs,perm,ro
root_rep_ckpt_53_2768_1 on /root_rep_ckpt_53_2768_1
ckpt,perm,ro
root_rep_ckpt_53_2768_2 on /root_rep_ckpt_53_2768_2
ckpt,perm,ro
```

---

**Note:** You can no longer write to the source file system.

---

From the destination side, type:

```
$ server_mount server_2
```

Destination output:

```
server_2 :
root_fs_2 on / uxfs,perm,rw
root_fs_common on /.etc_common uxfs,perm,ro
fs_ndmp2d on /fs_ndmp2d uxfs,perm,rw
fs2 on /fs2_replica1 uxfs,perm,rw
fs1_replica1 on /fs1_replica1 uxfs,perm,rw
root_rep_ckpt_72_2083_1 on /root_rep_ckpt_72_2083_1
ckpt,perm,ro
root_rep_ckpt_72_2083_2 on /root_rep_ckpt_72_2083_2
ckpt,perm,ro
```

---

**Note:** You can now write to the destination file system.

---

- To copy changes made on the secondary system back to the primary, use this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>} -reverse
-overwrite_destination
```

where:

<name> = name of the replication session to start

<session\_id> = ID of the replication session to start

Example:

To start replication session fs1\_1213, from the source side, type:

```
$ nas_replicate -start fs1_1213 -reverse -overwrite_destination
```

---

**Note:** This will reverse the direction of the replication to match the new mount status of the source and destination objects.

---

- Optionally, to return to the original replication direction (primary system to secondary system), use this command syntax:

```
$ nas_replicate -reverse {<name>|id=<session_id>}
```

where:

<name> = name of the replication session to start

<session\_id> = ID of the replication session to start

Example:

To start replication session fs1\_1213, type:

```
$ nas_replicate -reverse fs1_1213
```

---

## Fail over a replication session

You should perform a replication failover from the destination side only and when the source side is unavailable. The failover operation stops any data transfer in process and marks the destination object as read/write so that it can serve as the new source object.



**CAUTION** This operation is executed asynchronously and results in data loss if all the data was not transferred to the destination side prior to issuing this command.

Before you execute this command, note the following:

- ◆ EMC supports only one failed-over replication session per source object. If you fail over multiple sessions for a given source object, you can only save the changes from one of the failed-over replication sessions.
- ◆ Ensure that hosts have access to the destination Data Mover, so that they can access the destination object immediately.
- ◆ Failover operations do not change the direction of the replication.

1. Perform a failover by using this command syntax:

```
$ nas_replicate -failover {<name>|id=<session_id>}
```

where:

<name> = name of the replication session to failover

<session\_id> = ID of the replication session to failover

Example:

To perform a failover of replication session fs1\_12\_13, from the destination side, type:

```
$ nas_replicate -failover fs1_12_13
```

Output:

```
OK
```

**Note:** Execute this command from the Control Station on the destination VNX only.

2. Verify the failover from the destination side by using this command syntax:

```
$ nas_replicate -list
```

Output:

```
Name      type      Local Mover  Interconnect  Celerra  Status
fs1_12_13 filesystem server_2    <--20003    eng25212  Info
26045317429: Replication session is stopped.
Stopped replication session keeps replication configuration
information and internal snaps.
```

3. Optionally, verify the new mount status of the destination object by using this command syntax:

```
$ server_mount <movername>
```

where:

<movername> = the Data Mover where the file system is mounted

Example:

To display the mounted file systems on Data Mover server\_2 and verify that the destination file system fs2 is mounted read/write, from the destination side, type:

```
$ server_mount server_2
```

Output on destination:

```
server_2 :

root_fs_2 on / udfs,perm,rw
root_fs_common on /.etc_common udfs,perm,ro
fs1 on /fs1 udfs,perm,rw
fs_ndmp2d on /fs_ndmp2d udfs,perm,rw
fs2 on /fs2_replica1 udfs,perm,rw
root_rep_ckpt_59_1946_1 on /root_rep_ckpt_59_1946_1 ckpt,perm,ro
root_rep_ckpt_59_1946_2 on /root_rep_ckpt_59_1946_2 ckpt,perm,ro
```

Source output (when available):

```
server_2 :
root_fs_2 on / udfs,perm,rw
root_fs_common on /.etc_common udfs,perm,ro
fs1 on /fs1 udfs,perm,ro
fs1_ckpt1 on /fs1_ckpt1 ckpt,perm,ro
fs2 on /fs2 udfs,perm,ro
fs2_ckpt1 on /fs2_ckpt1 ckpt,perm,ro
root_rep_ckpt_59_2635_1 on /root_rep_ckpt_59_2635_1 ckpt,perm,ro
root_rep_ckpt_59_2635_2 on /root_rep_ckpt_59_2635_2 ckpt,perm,ro
```

4. When the source side becomes available, use the `nas_replicate -start` command with the `-reverse` option to start copying changes made on the secondary system back to the primary. Use this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>} -reverse
-overwrite_destination
```

where:

<name> = name of the replication session to start

<session\_id> = ID of the replication session to start

Example:

To start replication session fs1\_1213, type:

```
$ nas_replicate -start fs1_1213 -reverse -overwrite_destination
```

5. Optionally, to return to the original replication direction (primary system to secondary system), use this command syntax:

```
$ nas_replicate -reverse {<name>|id=<session_id>}
```

where:

<name> = name of the replication session to reverse

<session\_id> = ID of the replication session to reverse

Example:

To reverse replication session fs1\_1213, type:

```
$ nas_replicate -reverse fs1_1213
```



## Managing replication tasks

The tasks to manage replication tasks are:

- ◆ [Monitor replication tasks on page 150](#)
- ◆ [Abort a replication task on page 152](#)
- ◆ [Delete a replication task on page 156](#)

## Monitor replication tasks

Use the `nas_task` command to monitor long-running tasks, such as a one-time copy in progress and to find information about the progress or status of a replication command. You can also use `nas_task` to abort or delete replication tasks from the source or destination side of a replication session.

All replication commands are executed as a task in either synchronous or asynchronous mode. When in synchronous mode (default), the system frequently reports the progress and status of the completion of a command. Asynchronous or background mode is invoked immediately when the `-background` option is specified for a command.

A replication task can be in one of four states on a given VNX:

- ◆ Running — Task is successfully running
- ◆ Recovering — Task failed but has not yet completed
- ◆ Succeeded — Task completed successfully
- ◆ Failed — Task completed with errors

Each task is assigned a unique, numeric task ID per VNX which you can use to monitor the status of long-running tasks.

Use the `nas_task -list` command to list all running and completed tasks on a system.

Action						
To list all local tasks that are in progress, or completed tasks that have not been deleted, use this command syntax:						
<code>\$ nas_task -list</code>						
Output						
ID	Task State	Originator	Start Time	Description	Schedule	
4919	Running	root@cli.loc+	Fri Jan 11 07:46:04 EST 2008	Create Replication		
	f 3-copy-41.	Local System				

You can filter the task listing by task status by using the UNIX `grep` utility.

Action
To use the UNIX <code>grep</code> utility to list only tasks with a specific task status, use this command syntax:
<code>\$ nas_task -list  grep &lt;task status&gt;</code>
where:
<code>&lt;task status&gt;</code> = Status of the replication task: Running, Recovering, Succeeded, Failed
Example:
To list all tasks with a Running status on a system, type:

Action
<pre>\$ nas_task -list  grep Running</pre>
<p><b>Note:</b> Task status is case-sensitive.</p>
Output
<pre>1844 Running nasadmin@cli+ Tue Jan 22 07:54:35 EST 2008 Create Replication lp_fs2. eng19212</pre>

Use the `nas_task -info` option to obtain details about a task that was initiated on the local system.

Action
<p>To get information about a replication task on the system that initiated the task, use this command syntax from the source site:</p> <pre>\$ nas_task -info &lt;taskId&gt;</pre> <p>where:</p> <p>&lt;taskId&gt; = the unique task ID</p> <p>Example:</p> <p>To get detailed information about task 1844, type:</p> <pre>\$ nas_task -info 1844</pre>
Output
<pre>Task Id = 1844 Celerra Network Server = Local System Task State = Running Current Activity = Info 26045317613: Transfer done Movers = server_2 Percent Complete = 96 Description = Create Replication lp_fs2. Originator = nasadmin@cli.localhost Start Time = Tue Jan 22 07:54:35 EST 2008 Estimated End Time = Wed Jan 23 16:52:39 EST 2008 Schedule = n/a</pre>

You can also use the `-info` option to get information on a task that is running locally but was initiated from a remote VNX.

Action
<p>To get information about a replication task running on the local system but initiated from a remote system, use this command syntax from the source site:</p> <pre>\$ nas_task -info &lt;taskId&gt; -remote system &lt;remoteSystemName&gt;</pre> <p>where:</p> <p>&lt;taskId&gt; = the unique task ID</p>

**Action**

`<remoteSystemName>` = name of the remote VNX from which the task was initiated

Example:

To get information about task running on the local system but initiated from remote VNX cs110, type:

```
$ nas_task -info 1234 -remote system cs110
```

## Abort a replication task

You can abort a long-running task, such as a one-time copy in progress, from the source and destination sides of a replication. If you do, the VNX system attempts to return to the state it was in before the task started by undoing the completed parts of the task. Some operations cannot be aborted because they have reached a state where they cannot be undone. In this case, the system ignores the abort request. After you issue the abort command, you can monitor the progress of the abort. [Monitor replication tasks on page 150](#) provides details on monitoring replication tasks.

---

**Note:** Abort a task locally when a task is hung or you are experiencing network problems that cannot be resolved. You should attempt to fix any network problems before aborting a task.

---

### Full abort

Use the `nas_task -abort` command from the source side of a remote replication to abort a task running on the source and destination. This is a full abort and leaves the system in a consistent state. However, a full abort may take a long time to complete.

To perform a full abort of a replication task:

1. Display the task ID for all the local tasks that are in progress, by using this command syntax:

```
$ nas_task -list
```

---

**Note:** You need the task ID to get task information about the task to abort.

---

Output:

```
ID      Task State Originator   Start Time  Description  Schedule
Remote System
1637 Running   nasadmin@cli+ Fri Jan 25 14:47:44 EST 2008 Create
Replication fs1_local.
cs100
```

2. Get information about the replication task on the system that initiated the task by using this command syntax:

```
$ nas_task -info <taskId>
```

where:



`<taskId>` = unique task ID

Example:

To get detailed information about task 1637, type:

```
$ nas_task -info 1637
```

Output:

```
Task Id                = 1637
Celerra Network Server = cs100
Task State              = Running
Current Activity        = Info 26045317581: Creating session
Movers                  = server_2,server_3
Percent Complete        = 94
Description             = Create Replication fs1_local.
Originator              = nasadmin@cli.localhost
Start Time              = Fri Jan 25 14:47:44 EST 2008
Estimated End Time     = Fri Jan 25 13:55:05 EST 2008
Schedule                = n/a
```

3. Abort the replication task on the system that initiated the task by using this command syntax:

```
$ nas_task -abort <taskId>
```

where:

`<taskId>` = ID of the task to abort

Example:

To abort task 1637, type:

```
$ nas_task -abort 1637
```

Output:

```
OK
```

### Local abort

You can also abort a task on a specific local Data Mover. However, you should perform a local abort with caution as the entire system will be left in an inconsistent state until the task is aborted on all Data Movers running the task.

 **CAUTION** Use caution when using this option.

You must perform a local abort on both Data Movers on which the task is running.

Follow this procedure to abort a task on the VNX system that initiated the task:

1. Display the task ID for all the local tasks that are in progress by using this command syntax:

```
$ nas_task -list
```

You need the task ID to get task information about the task to abort.

Output:

```

ID    Task State Originator Start Time Description Schedule Remote System
1689  Running    nasadmin@cli+ Fri Jan 25 14:55:00 EST 2008 Create
Replication fs2_local.                cs100
1637  Failed     nasadmin@cli+ Fri Jan 25 14:47:44 EST 2008 Create
Replication fs1_local.                cs100
1531  Succeed    nasadmin@cli+ Fri Jan 25 14:18:30 EST 2008 Create
Replication repl_v2.                  cs100

```

2. Get information about the replication task on the system that initiated the task by using this command syntax:

```
$ nas_task -info <taskId>
```

where:

<taskId> = the unique task ID

Example:

To get detailed information about task 1689, type:

```
$ nas_task -info 1689
```

Output:

```

Task Id                = 1689
Celerra Network Server = cs100
Task State             = Running
Current Activity       = Info 26045317581: Creating session
Movers                 = server_2,server_3
Percent Complete       = 94
Description            = Create Replication fs2_local.
Originator             = nasadmin@cli.localhost
Start Time             = Fri Jan 25 14:55:00 EST 2008
Estimated End Time    = Fri Jan 25 14:02:55 EST 2008
Schedule               = n/a

```

3. Abort a replication task running locally on the VNX that initiated the task by using this command syntax:

```
$ nas_task -abort <taskId> -mover <moverName>
```

where:

<taskId> = ID of the task to abort

<moverName> = Data Mover on which the task is running

Example:

To abort task 1689 running locally on Data Mover server\_2, type:

```
$ nas_task -abort 1689 -mover server_2
```

---

**Note:** Run this command for each Data Mover on which the task is running. The system will be in an inconsistent state until the task is aborted on all Data Movers running the task.

---

Output:

```
OK
```

- Optionally, verify that the task is aborting and in Recovering state by using this command syntax:

```
$ nas_task -list
```

Output:

```
ID      Task State Originator  Start Time  Description  Schedule
Remote System
1689 Recovering nasadmin@cli+ Fri Jan 25 14:55:00 EST 2008 Create
Replication fs2_local. cs100
1637 Failed      nasadmin@cli+ Fri Jan 25 14:47:44 EST 2008 Create
Replication fs1_local. cs100
1531 Failed      nasadmin@cli+ Fri Jan 25 14:18:30 EST 2008 Create
Replication repl_v2.   cs100
```

This procedure aborts a local task initiated by a remote VNX:

- Display the task ID for the task to abort by using this command syntax:

```
$ nas_task -list
```

Output:

```
ID      Task State Originator  Start Time  Description  Schedule Remote
System
3833 Running   nasadmin@cli+ Wed Jan 30 15:15:13 EST 2008 Create
Replication rem1_v2.
cs110
```

- Get information about the task on the system that initiated the task by using this command syntax:

```
$ nas_task -info <taskId>
```

where:

<taskId> = the unique task ID

Example:

To get detailed information about task 3883 originated from VNX cs110, type:

```
$ nas_task -info 3833 -remote_system cs110
```

Output:

```
Task Id                = 3833
Celerra Network Server = cs110
Task State              = Running
Current Activity        = Info 26045317587: Cleaning up
Movers                  = server_2
Description             = Create Replication rem1_v2.
Originator              = nasadmin@cli.localhost
Start Time              = Wed Jan 30 15:15:13 EST 2008
Estimated End Time     = Wed Dec 31 19:00:00 EST 1969
Schedule                = n/a
```

- To abort a replication task initiated by a remote VNX and running locally, use this command syntax:

```
$ nas_task -abort <taskId> -mover <moverName> -remote_system
```

```
{<remoteSystemName>|id=<id>}
```

where:

<taskId> = ID of the task to abort

<moverName> = Data Mover on which the task is running

<remoteSystemName> = name of the remote VNX system

<id> = ID of the remote VNX system

Example:

To abort task 3833 running on Data Mover server \_2 and initiated by remote VNX system cs110, type:

```
$ nas_task -abort 3833 -mover server_2 -remote_system cs110
```

Output:

```
Info 26307199024: Abort request sent. Please check the task status to see if task was successfully aborted.
```

4. Check the task status for task 3833 by using this command syntax:

```
$ nas_task -info <taskId> -remote_system {<remoteSystemName>|id=<id>}
```

where:

<taskId> = ID of the task to abort

<remoteSystemName> = name of the remote VNX system

<id> = ID of the remote VNX system

Example:

To check the status of task 3833, type:

```
$ nas_task -info 3833 -remote_system cs110
```

Output:

```
Error 13422297148: Task cs110:3833 not found.
```

## Delete a replication task

After an asynchronous task completes, you can manually delete the task by using the `nas_task -delete` option, or you can wait 3 days for the system to delete the task automatically. You run the `nas_task delete` command from the system that initiated the replication.

1. Obtain the task ID for the task you want to delete, by using this command syntax:

```
$ nas_task -list
```

---

**Note:** You need the task ID to delete a task.

---

Output:

ID	Task State	Originator	Start Time	Description	Schedule	Remote System
1689	Failed	nasadmin@cli+	Fri Jan 25 14:55:00	EST 2008	Create	Replication fs2_local. eng17310
1637	Failed	nasadmin@cli+	Fri Jan 25 14:47:44	EST 2008	Create	Replication fs1_local. eng17310
1531	Failed	nasadmin@cli+	Fri Jan 25 14:18:30	EST 2008	Create	Replication repl_v2. eng17310

2. Delete a completed task from the system that initiated the task, by using this command syntax:

```
$ nas_task -delete <taskId>
```

where:

<taskId> = ID of the task to delete

Example:

To delete task 1531, type:

```
$ nas_task -delete 1531
```

Output:

OK



## Managing Data Mover interconnects

After you have established a Data Mover interconnect for replication sessions to use, you can manage the interconnect.

The tasks to manage Data Mover interconnects are:

- ◆ [View a list of Data Mover interconnects on page 160](#)
- ◆ [View Data Mover interconnect information on page 160](#)
- ◆ [Modify Data Mover interconnect properties on page 161](#)
- ◆ [Change the interfaces associated with an interconnect on page 163](#)
- ◆ [Pause a Data Mover interconnect on page 168](#)
- ◆ [Resume a paused Data Mover interconnect on page 169](#)
- ◆ [Validate a Data Mover interconnect on page 169](#)
- ◆ [Delete a Data Mover interconnect on page 170](#)

## View a list of Data Mover interconnects

You can list all interconnects available on the local VNX, and optionally list the interconnects on the local VNX that have peer interconnects on a specified destination VNX.

Action																					
<p>To view a list of Data Mover interconnects on the local VNX cabinet, and optionally, the interconnects on a remote cabinet, use this command syntax:</p> <pre>\$ nas_cel -interconnect -list [-destination_system {&lt;cel_name&gt; id=&lt;cel_id&gt;}]</pre> <p>where:</p> <p>&lt;cel_name&gt; = name of a remote (destination) VNX</p> <p>&lt;cel_id&gt; = ID of a remote VNX</p> <p>Example:</p> <p>To list the interconnects established on local VNX site NY, type:</p> <pre>\$ nas_cel -interconnect -list</pre>																					
Output																					
<table border="1"> <thead> <tr> <th>id</th> <th>name</th> <th>source_server</th> <th>destination_system</th> <th>destination_server</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>NYs2_LAs2</td> <td>server_2</td> <td>celerra_5</td> <td>server_2</td> </tr> <tr> <td>15</td> <td>NYs2_LAs3</td> <td>server_2</td> <td>celerra_6</td> <td>server_3</td> </tr> <tr> <td>20</td> <td>loopback</td> <td>server_2</td> <td>celerra_5</td> <td>server_2</td> </tr> </tbody> </table> <p><b>Note:</b> All loopback interconnects display "loopback" for the interconnect name. Use the interconnect ID to specify a specific loopback interconnect.</p>		id	name	source_server	destination_system	destination_server	1	NYs2_LAs2	server_2	celerra_5	server_2	15	NYs2_LAs3	server_2	celerra_6	server_3	20	loopback	server_2	celerra_5	server_2
id	name	source_server	destination_system	destination_server																	
1	NYs2_LAs2	server_2	celerra_5	server_2																	
15	NYs2_LAs3	server_2	celerra_6	server_3																	
20	loopback	server_2	celerra_5	server_2																	

## View Data Mover interconnect information

You can view interconnect information for a specific interconnect (identified by name or ID), or you can view information for all interconnects.

Action	
<p>To display detailed information about a Data Mover interconnect, use this command syntax:</p> <pre>\$ nas_cel -interconnect -info {&lt;name&gt; id=&lt;interConnectId&gt; -all}</pre> <p>where:</p> <p>&lt;name&gt; = name of a Data Mover interconnect</p> <p>&lt;interConnectId&gt; = ID of Data Mover interconnect</p> <p>Example:</p> <p>To view detailed information about interconnect NYs2_LAs2 established on the local VNX site NY, type:</p> <pre>\$ nas_cel -interconnect -info NYs2_LAs2</pre>	



**Output**

```

id                = 1
name              = NYs2_LAs2
source_server     = server_2
source_interfaces = 45.252.2.3
destination_system = celerra_5
destination_server = server_2
destination_interfaces = 45.252.2.5
bandwidth schedule =
crc enabled       = yes
number of configured replications = 0
number of replications in transfer = 0
status           = The interconnect is OK.

```

## Modify Data Mover interconnect properties

You can modify the interconnect name, the list of interfaces available for use on the local or peer side of the interconnect, and the bandwidth schedule.

You cannot modify:

- ◆ A loopback interconnect.
- ◆ An interface in use by a replication session.
- ◆ The peer side of an interconnect configured on a remote system. You must modify it from that system.

Before you make changes to an interconnect, review [Interconnect setup considerations on page 58](#).

**Action**

Modify the name of a Data Mover interconnect, using this command syntax:

```
$ nas_cel -interconnect -modify {<name>|id=<interConnectId>} -name <newName>
```

where:

<name> = name of the interconnect

<interConnectId> = ID of the interconnect

<newName> = new name for the interconnect

Example:

To rename interconnect ny2\_ny3 to s2\_s3, type:

```
$ nas_cel -interconnect -modify ny2_ny3 -name s2_s3
```

**Output**

```
operation in progress (not interruptible)...
id = 30006
name = NY-to-LA
source_server = server_3
source_interfaces = 45.252.2.4
destination_system = eng25212
destination_server = server_3
destination_interfaces = 45.252.0.71
bandwidth schedule = use available bandwidth
crc enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.
```

This is an example of how to update the source interface list.

**Action**

To modify the list of Data Mover IP addresses available for use on the local side of the interconnect, use this command syntax:

```
$ nas_cel -interconnect -modify {<name>|id=<interConnectId>}
-source_interfaces {<name_service_interface_name>|ip=<ipaddr>},...
-destination_interfaces {<name_service_interface_name>|ip=<ipaddr>},...
```

where:

<name> = name of the interconnect

<interConnectId> = ID of the interconnect

<name\_service\_interface\_name> = interface defined using a name service interface name that must resolve to a single IP address

<ipaddr> = interface defined by an IP address

Example:

To modify the list of source interfaces available on interconnect s2\_s3, type:

```
$ nas_cel -interconnect -modify s2_s3 -source_interfaces ip=172.24.102.0,
ip=172.24.103,ip=172.24.104
```

**Note:** To avoid problems with interface selection, any changes made to the interface lists should be reflected on both sides of an interconnect.

**Output**

OK

This is an example of how to update the bandwidth schedule for a Data Mover interconnect.

**Action**

To modify the bandwidth schedule for a Data Mover interconnect, use this syntax:

Action
<pre>\$ nas_cel -interconnect -modify {&lt;name&gt; id=&lt;interConnectId&gt;} -bandwidth &lt;bandwidth&gt;</pre> <p>where:</p> <p>&lt;name&gt; = name of the interconnect</p> <p>&lt;interConnectId&gt; = ID of the interconnect</p> <p>&lt;bandwidth&gt; = schedule with one or more comma-separated entries, most specific to least specific as follows:  [[Su Mo Tu We Th Fr Sa]][HH:00-HH:00]/[Kbps]</p> <p>,[ &lt;next_entry&gt;],[...]</p> <p>Example:</p> <p>To modify the bandwidth schedule for interconnect s2_s3 and set bandwidth limits to 2000 Kb/s from 7 A.M. to 6 P.M. Monday through Friday, otherwise use 8000 Kb/s, type:</p> <pre>\$ nas_cel -interconnect -modify s2_s3 -bandwidth MoTuWeThFr07:00-18:00/2000,/8000</pre>
Output
OK

## Change the interfaces associated with an interconnect

Before you change the interface that is being used by an existing Data Mover interconnect, you must first stop the replication session, make the modification to the interconnect, and then start the session again.

This is an example of how to change the interfaces associated with a Data Mover interconnect. In this example, there are two interfaces on the source VNX (cge2 and cge0) and one interface on the destination VNX (cge0).

We will create Data Mover interconnects and then a file system replication that uses device cge0 with source interface IP 45.252.1.42 and device cge0 with destination interface IP 45.252.1.46. Then, we will stop the replication to modify the Data Mover interconnects to use source device cge2, instead of cge0 and restart the replication.

1. To display information for all interfaces on a Data Mover on the source VNX, type:

```
$ server_ifconfig server_3 -all
```

Output:

```

server_3 :

cge2_52 protocol=IP device=cge2
    inet=45.252.1.52 netmask=255.255.255.0
broadcast=45.252.1.255
    UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:1f:8a:b6
loop protocol=IP device=loop
    inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
    UP, loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0
netname=localhost
cge0_3 protocol=IP device=cge0
    inet=45.252.1.42 netmask=255.255.255.0
broadcast=45.252.1.255
    ...

```

2. To display information for all interfaces on a Data Mover on the destination VNX, type:

```
$ server_ifconfig server_3 -all
```

Output:

```

server_3 :
loop protocol=IP device=loop
    inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
    UP, loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0
netname=localhost
cge0-46 protocol=IP device=cge0
    inet=45.252.1.46 netmask=255.255.255.0
broadcast=45.252.1.255
    UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:25:fa:e
    el3l protocol=IP device=mge1
    inet=128.221.253.3 netmask=255.255.255.0
broadcast=128.221.253.255
    UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:25:49:b3
netname=localhost
    ...

```

3. To create the local interconnect 40s3-41s3 from the source to the destination VNX, type:

```

$ nas_cel -interconnect -create 40s3-41s3 -source_server server_3
  -destination_system eng25241 -destination_server server_3 -source_interfaces
  ip=45.252.1.42 -destination_interfaces ip=45.252.1.46

```

Output:

```

operation in progress (not interruptible)...
id = 30003
name = 40s3-41s3
source_server = server_3
source_interfaces = 45.252.1.42
destination_system = eng25241
destination_server = server_3
destination_interfaces = 45.252.1.46
bandwidth schedule = uses available bandwidth
crc enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.

```

4. To create the peer interconnect 41s3-40s3 from the destination to the source, type:

```
$ nas_cel -interconnect -create 41s3-40s3 -source_server server_3
  -destination_system eng25240 -destination_server server_3 -source_interfaces
  ip=45.252.1.46 -destination_interfaces ip=45.252.1.42
```

Output:

```
operation in progress (not interruptible)...
id = 30005
name = 41s3-40s3
source_server = server_3
source_interfaces = 45.252.1.46
destination_system = eng25240
destination_server = server_3
destination_interfaces = 45.252.1.42
bandwidth_schedule = uses available bandwidth
crc_enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.
```

5. To create replication session ps1-40-41 to replicate file system ps1 from the source to the destination, type:

```
$ nas_replicate -create ps1-40-41 -source -fs ps1 -destination -pool id=3
  -interconnect 40s3-41s3
```

Output:

```
OK
```

6. To display information for replication ps1-40-41 on the source VNX, type:

```
$ nas_replicate -info ps1-40-41
```

Output:

```

ID = 844_APM00083201400_0000_1012_APM00083400465_0000
Name = ps1-40-41
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Tue Apr 21 07:44:27 EDT 2009
Type = filesystem
Celerra Network Server = eng25241
Dart Interconnect = 40s3-41s3
Peer Dart Interconnect = 41s3-40s3
Replication Role = source
Source Filesystem = ps1
Source Data Mover = server_3
Source Interface = 45.252.1.42
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = ps1
Destination Data Mover = server_3
Destination Interface = 45.252.1.46
Destination Control Port = 5085
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 10
Next Transfer Size (KB) = 8664
Current Transfer Size (KB) = 8664
Current Transfer Remain (KB) = 86
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 35219
Current Read Rate (KB/s) = 11581
Current Write Rate (KB/s) = 3528
Previous Transfer Rate (KB/s) = 2801
Previous Read Rate (KB/s) = 6230
Previous Write Rate (KB/s) = 3528
Average Transfer Rate (KB/s) = 2801
Average Read Rate (KB/s) = 6230
Average Write Rate (KB/s) = 1834

```

- To change the interface on the Data Mover to use cge2, instead of cge0, stop the replication session at the source VNX by typing:

```
# nas_replicate -stop ps1-40-41 -mode both
```

Output:

```
OK
```

- To modify Data Mover interconnect 40s3-41s3 to use device cge2, which uses IP 45.252.1.52, type:

```
$ nas_cel -interconnect -modify 40s3-41s3 -source_interfaces ip=45.252.1.52
```

Output:

```

operation in progress (not interruptible)...
id = 30003
name = 40s3-41s3
source_server = server_3
source_interfaces = 45.252.1.52
destination_system = eng25241
destination_server = server_3
destination_interfaces = 45.252.1.46
bandwidth schedule = uses available bandwidth
crc enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.

```

9. To modify peer Data Mover interconnect 41s3-40s3 to use device cge2, type:

```
$ nas_cel -interconnect -modify 41s3-40s3 -destination_interfaces ip=45.252.1.52
```

Output:

```

operation in progress (not interruptible)...
id = 30005
name = 41s3-40s3
source_server = server_3
source_interfaces = 45.252.1.46
destination_system = eng25240
destination_server = server_3
destination_interfaces = 45.252.1.52
bandwidth schedule = uses available bandwidth
crc enabled = yes
number of configured replications = 0
number of replications in transfer = 0
status = The interconnect is OK.

```

10. Optionally, to validate the Data Mover interconnect, type:

```
$ nas_cel -interconnect -validate 40s3-41s3
```

Output:

```
40s3-41s3: validating 1 interface pairs; please wait ... ok
```

11. To start the replication session ps1-40-41 and specify the Data Mover Interconnect, type:

```
$ nas_replicate -start ps1-40-41 -interconnect 40s3-41s3
```

Output:

```
OK
```

12. To display information for replication ps1-40-41 on the source VNX and see that the source interface has changed, type:

```
$ nas_replicate -info ps1-40-41
```

Output:

```

ID = 844_APM00083201400_0000_1012_APM00083400465_0000 Name = ps1-40-41
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Tue Apr 21 07:52:51 EDT 2009
Type = filesystem
Celerra Network Server = eng25241
Dart Interconnect = 40s3-41s3
Peer Dart Interconnect = 41s3-40s3
Replication Role = source
Source Filesystem = ps1
Source Data Mover = server_3
Source Interface = 45.252.1.52
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = ps1
Destination Data Mover = server_3
Destination Interface = 45.252.1.46
Destination Control Port = 5085
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 10
Next Transfer Size (KB) = 0
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 0
Current Read Rate (KB/s) = 0
Current Write Rate (KB/s) = 0
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 0
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0
    
```

## Pause a Data Mover interconnect

When you pause a Data Mover interconnect, it temporarily stops all data transmission for all replication sessions that are using the specified interconnect. Data transmission remains paused until you either resume transmission over the interconnect or delete the interconnect.

**Note:** Setting the throttle bandwidth value to 0 will also suspend data transfer.

### Action

To pause a Data Mover interconnect, use this command syntax:

```
$ nas_cel -interconnect -pause {<name>|id=<interConnectId>}
```

where:

<name> = name of the interconnect

<interConnectId> = ID of the interconnect

Example:



Action
To pause the Data Mover s3_s4, type:  \$ <b>nas_cel -interconnect -pause s3_s4</b>
Output
done

## Resume a paused Data Mover interconnect

Use the `-resume` command to resume data transmission over a Data Mover interconnect that has been paused.

Action
To resume a paused Data Mover interconnect, use this command syntax:  \$ <b>nas_cel -interconnect -resume</b> {<name> id=<interConnectId>}  where:  <name> = name of the Data Mover interconnect <interConnectId> = ID of the Data Mover interconnect  Example:  To resume data transmission over Data Mover s3_s4, type:  \$ <b>nas_cel -interconnect -resume s3_s4</b>
Output
done

## Validate a Data Mover interconnect

You can verify that authentication is configured properly between a given Data Mover pair and validate all the combinations between source and destination IP addresses/interface name.

Action
To validate a Data Mover interconnect, use this command syntax:  \$ <b>nas_cel -interconnect -validate</b> {<name> id=<interConnectId>}  where:  <name> = name of the Data Mover interconnect <interConnectId> = ID of the Data Mover interconnect  Example:  To validate Data Mover s3_s4, type:

<b>Action</b>
<code>\$ nas_cel -interconnect -validate s3_s4</code>
<b>Output</b>
validating...ok.

## Delete a Data Mover interconnect

Follow this procedure to delete a Data Mover interconnect.

You cannot delete an interconnect that is in use by a replication session.

<b>Action</b>
<p>To delete a Data Mover interconnect, use this command syntax:</p> <pre>\$ nas_cel -interconnect -delete {&lt;name&gt; id=&lt;interConnectId&gt;}</pre> <p>where:</p> <p>&lt;name&gt; = name of the Data Mover interconnect to delete</p> <p>&lt;interConnectId&gt; = ID of the Data Mover interconnect to delete</p> <p>Example:</p> <p>To delete Data Mover interconnect s3_s4, type:</p> <pre>\$ nas_cel -interconnect -delete s3_s4</pre>
<b>Output</b>
done

## Managing the replication environment

The tasks to manage the VNX Replicator environment are:

- ◆ [Monitor replication on page 172](#)
- ◆ [View the passphrase on page 172](#)
- ◆ [Change the passphrase on page 173](#)
- ◆ [Extend the size of a replicated file system on page 174](#)
- ◆ [Change the percentage of system space allotted to SavVols on page 174](#)
- ◆ [Change replication parameters on page 175](#)
- ◆ [Mount the source or destination file system on a different Data Mover on page 176](#)
- ◆ [Rename a Data Mover that has existing interconnects on page 176](#)
- ◆ [Change the mount status of source or destination object on page 177](#)
- ◆ [Perform an initial copy by using the disk transport method on page 178](#)
- ◆ [Change Control Station IP addresses on page 180](#)
- ◆ [Perform an initial copy by using the tape transport method on page 183](#)
- ◆ [Recover from a corrupted file system using nas\\_fsck on page 185](#)
- ◆ [Manage expected outages on page 186](#)
- ◆ [Manage unexpected outages on page 188](#)

## Monitor replication

Table 8 on page 172 shows the commands to use to monitor different aspects of replication. You can also monitor replication by using Unisphere, which is described in the Unisphere online help.

**Table 8. Ways to monitor replication**

Monitor	Description	Command
Data Movers	Returns information about available memory and the CPU idle percentage.	server_sysstat
File systems	Reports the amount of used and available disk space for a file system and reports the amount of a file system's total capacity that is used.	server_df
File systems	Reports the size of a filesystem and the checkpoints used.	nas_fs -list
File systems	Reports replication sessions that are stopped or configured on a file system.	nas_fs -info
Replication	Lists status information for all replication sessions or a specific session, as described in <a href="#">(Optional) Verify file system replication on page 85</a> .	nas_replicate -list
Replication	Shows detailed information for an individual replication session, as detailed in <a href="#">(Optional) Verify file system replication on page 85</a> .	nas_replicate -info
Replication	Lists task status information for an individual replication task or for all replication tasks, as described in <a href="#">Monitor replication tasks on page 150</a> .	nas_task -list
Replication	Shows detailed task information for an individual replication task or for all replication tasks, as described in <a href="#">Monitor replication tasks on page 150</a> .	nas_task -info

## View the passphrase

The passphrase is used to authenticate with a remote VNX system.

Action
<p>To view the passphrase of a remote VNX system, use this command syntax:</p> <pre>\$ nas_cel -info {&lt;cel_name&gt; id=&lt;cel_id&gt;}</pre> <p>where:</p> <p>&lt;cel_name&gt; = ID of the remote VNX system &lt;cel_id&gt; = name of the remote VNX system</p> <p>The VNX system ID is assigned automatically. To view this ID for a remote system, use the nas_cel -list command.</p> <p>Example:</p> <p>To view the passphrase of the VNX system, type:</p> <pre>\$ nas_cel -info id=5</pre>
Output
<pre>id          = 5 name        = cs110 owner       = 503 device      = channel     = net_path    = 192.168.168.102 celerra_id  = APM000446038450000 passphrase  = nasadmin</pre>

## Change the passphrase

Use this procedure to change the passphrase on a VNX system.

**Note:** The passphrase on both the local and remote systems must be identical.

1. At each site, review the passphrase, by using this command syntax:

```
$ nas_cel -info {<cel_name>|id=<cel_id>}
```

where:

<cel\_name> = name of the VNX system

2. At each site, modify the passphrase, by using this command syntax:

```
$ nas_cel -modify {<cel_name>|id=<cel_id>} -passphrase <passphrase>
```

where:

<cel\_name> = name of the VNX system

<cel\_id> = ID of the VNX system

<passphrase> = current common passphrase

---

## Extend the size of a replicated file system

During normal replication operation, the source and destination file systems must be the same size. If you want to extend the source file system, the destination file system must have enough storage available, so that it can be extended as well. Only the source file system can be extended by the administrator. The extension of the destination file system takes place automatically and cannot be initiated by the VNX administrator. When the source file system is extended by the administrator, the system will extend the source file system first. The destination file system will be extended upon the next data transfer. If there is an error during extension of the destination file system, then the replication data transfer will not proceed until the situation is corrected.

VNX Replicator can create a destination file system for your virtually provisioned source file systems. Since auto-extension is enabled for a virtually provisioned source file system, the destination file system need not be virtually provisioned. When the virtually provisioned source file system is automatically extended, VNX Replicator ensures that the destination file system is also extended by the same amount of disk space.

*Managing Volumes and File Systems with VNX Automatic Volume Management* provides more details about Automatic File System Extension.

### Recovering from replicated file system extension failure

An extension of a replicated source file system always extends the source file system first. If there is a failure during this extension, the administrator will know immediately about this failure. If the source file system extension fails, the destination file system will not be extended. If the source file system extension is successful, the next data transfer from the source to the destination triggers an automatic destination file system extension.

A replication destination file system employs an internal script that extends the file system to the required size. When an error occurs during the destination file system extension, the cause of the error is logged to the `sys_log` file. The administrator should view `sys_log` to find the exact error, and then use `nas_message -info` command to obtain the detailed description of the error and the recommended corrective actions.

---

## Change the percentage of system space allotted to SavVols

To prevent the SavVol from running out of space for file systems with high change rates or file systems with a high number of checkpoints, increase the percentage of system space allotted for SavVols. If you increase the SavVol space, you cannot decrease it. You can reclaim the space used by deleting all user checkpoints and replications on the system.

This setting applies to all SavVols for all checkpoints and not just VNX Replicator checkpoints.

---

**Note:** This setting will be overwritten by a code upgrade. Therefore, you will need to perform this action again.

---

Use this procedure to change the percentage of system space allotted to all SavVols.

1. Log in to the Control Station.
2. Open `/nas/sys/nas_param` with a text editor.  
A short list of configuration lines appears.
3. Locate this SnapSure configuration line in the file:

```
ckpt:10:100:20:
```

where:

`10` = Control Station polling interval rate in seconds

`100` = maximum rate at which a file system is written in MB/second

`20` = percentage of the entire system's volume allotted to the creation and extension of all the SavVols used by VNX

---

**Note:** If this line does not exist, it means the SavVol-space-allotment parameter is currently set to its default value of 20, which means 20 percent of the system space can be used for SavVols. To change this setting, you must first add the line: `ckpt:10:100:20:`

---

4. Change the third parameter, which is the percent of entire system's volume allotted to SavVols as needed. Values are the default 20 through 99.

---

**Note:** To ensure proper functionality, do not use a value below 20 percent for the third value.

---

Do not change the Control Station event polling interval rate, default =10, or the maximum rate to which a file system is written, default =100. Doing so will have a negative impact on system performance. Do not change any other lines in this file without a thorough knowledge of the potential effects on the system. Contact your EMC Customer Support Representative for more information.

5. Save and close the file.

---

**Note:** Changing this value does not require a Data Mover or Control Station restart.

---

The *Parameters Guide for VNX for File* provides more information on changing system parameters.

## Change replication parameters

If you choose to set these policies for all sessions on a Data Mover, you can change these parameters:

- ♦ `nas_param` — By default, the space used by SavVol cannot exceed 20 percent of the total space of the cabinet. You can increase the space by using the `nas_param` file.

- ♦ `server_param` — By default, `.ts` is the value used for the `NDMP tapeSilveringStr` parameter. You can specify a new value by using the `server_param` file.

The *Parameters Guide for VNX for File* describes all VNX system parameters.

## Mount the source or destination file system on a different Data Mover

After you stop a replication, if you choose to move your source or destination file system to another Data Mover, you must first unmount your file system and all associated checkpoints. Then, mount all the checkpoints and the file system on a different Data Mover.

To move your file system to another Data Mover:

1. Verify whether there is an interconnect set up between Data Movers. If not, create the interconnect.
2. Ensure there is no client access to the file system.
3. Unmount any user-defined checkpoints associated with the file system if they exist.
4. Unmount the file system. Internal checkpoints are automatically unmounted.
5. Mount the file system on a different Data Mover.
6. Mount all the user-defined checkpoints on the same Data Mover where your file system resides.
7. Start the replication session by using a new interconnect.

## Rename a Data Mover that has existing interconnects

After you configure an interconnect for a Data Mover, you cannot rename that Data Mover without deleting and reconfiguring the interconnect and reestablishing replication sessions.

This procedure describes how to rename a Data Mover that has one or more existing interconnects, other than a loopback interconnect, configured for it.

1. Stop all replications on the source and destination by using the Data Mover to be renamed. Follow the procedure, [Stop a replication session on page 134](#).
2. Delete all interconnects on the Data Mover. Follow the procedure, [Delete a Data Mover interconnect on page 170](#).
3. Rename the Data Mover, using this command syntax:

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

where:

<movername> = current hostname of the specified Data Mover

<new\_name> = new hostname for the Data Mover

Example:



To change the current hostname for a Data Mover, type:

```
$ server_name server_2 my_srv2
```

Output:

```
server_2 : my_srv2
```

4. Re-create all the interconnects on the newly named Data Mover. Follow the procedure, [Set up one side of an interconnect on page 74](#).
5. Delete all the peer interconnects on the remote VNX system. Follow the procedure, [Delete a Data Mover interconnect on page 170](#).
6. Re-create all the peer interconnects on the remote VNX system specifying the new Data Mover name. Follow the procedure, [Set up the peer side of the interconnect on page 76](#).
7. Validate the new interconnects and correct any configuration errors. Follow the procedure, [Validate interconnect information on page 77](#).
8. Start the replication sessions specifying the new interconnect name. Follow the procedure, [Start a replication session on page 136](#).

## Change the mount status of source or destination object

This procedure describes how to change the mount status of a replication source or destination object.

Use this procedure to change the mount status of a source object when starting one or more failed over or switched over replication sessions in a one-to-many configuration.

If you are starting a replication session from the destination side and the source object is used in only one replication session, you need not change the mount status of the destination object. However, if the source object is used in multiple replication sessions, you may need to change the mount status of each of the source and destination objects involved in all of the replication sessions in the configuration to start each session successfully.

**CAUTION** When you change the status of a replication object, you risk changing the contents of the object, which can result in data being overwritten.

### File system

To change the mount status of a file system, use the `server_mount` command.

Action
<p>To change the mount status of a file system, use this command syntax:</p> <pre>\$ server_mount &lt;movername&gt; -option [ro rw] &lt;fs_name&gt;</pre> <p>where:</p> <p>&lt;movername&gt; = name of the Data Mover where the file system resides</p>

Action
<p><b>ro rw</b> = the mount is read/write (default), or read-only</p> <p><b>&lt;fs_name&gt;</b> = name of the file system</p> <p>Example:</p> <p>To mount file system fs1 with read-only access, type:</p> <pre>\$ server_mount server_2 -option ro fs1</pre>
Output
<pre>server_2 : done</pre>

### VDM

To change the state of a VDM, use the `nas_server` command.

Action
<p>To change the state of a VDM, use this command syntax:</p> <pre>\$ nas_server -vdm &lt;vdm_name&gt; -setstate &lt;state&gt;</pre> <p>where:</p> <p><b>&lt;vdm_name&gt;</b> = name of the VDM whose state you want to change</p> <p><b>&lt;state&gt;</b> = VDM state, mounted for read-only or loaded for read/write</p> <p>Example:</p> <p>To set the state of vdm_1 from loaded to mounted, type:</p> <pre>\$ nas_server -vdm vdm_1 -setstate mounted</pre>
Output
<pre>id = 3 name = vdm_1 acl = 0 type = vdm server = server_2 rootfs = root_fs_vdm_1 I18N mode = UNICODE mountedfs = member_of = status : defined = enabled actual = mounted Interfaces to services mapping:</pre>

## Perform an initial copy by using the disk transport method

Follow this procedure to perform an initial copy of the file system by using the disk transport method. [Performing an initial copy by using disk or tape on page 54](#) describes how the disk transport method works:

1. Create a checkpoint of the source file system on VNX system A, by using this command syntax:

```
$ fs_ckpt <fsName> -name <ckptName> -create
```

where:

<fsName> = name of the source file system

<ckptName> = name of the source file system checkpoint

Example:

To create checkpoint ufs1\_ckpt1 on VNX system A, type:

```
$ fs_ckpt ufs1 -name ufs1_ckpt1 -Create
```

2. Create a one-time copy session to copy the source file system checkpoint from VNX system A to VNX system B, by using this command syntax:

```
$ nas_copy -name <name> -source -ckpt <ckptName> -destination -pool
{id= <dstStoragePoolId>|<dstStoragePool>} -interconnect
{<name>|id=<interConnectId>}
```

where:

<name> = name of the copy session

<ckptName> = name of the existing source checkpoint to copy

<dstStoragePoolId> = ID of the storage pool to use to create the checkpoint on the destination

<dstStoragePool> = name of the storage pool to use to create the checkpoint on the destination

<name>= name of the Data Mover interconnect to use for this copy session

<interConnectId> = ID of the Data Mover interconnect to use for this copy session

Example:

To copy checkpoint ufs1\_ckpt1 to VNX system B, type:

```
$ nas_copy -name rep1 -source -ckpt ufs1_ckpt1 -destination -pool id=3
-interconnect id=20005
```

After the copy, file system ufs1 and checkpoint ufs1\_ckpt1 are created on VNX system B.

3. Move VNX system B to the destination location.
4. Create a one-time copy session to copy the checkpoint from VNX system B to VNX system C by using this command syntax:

```
$ nas_copy -name <name> -source -ckpt <ckptName> -destination -pool
{id=<dstStoragePoolId>|<dstStoragePool>} -interconnect
{<name>|id=<interConnectId>}
```

where:

*<name>* = name of the copy session

*<ckptName>* = name of the existing source checkpoint to copy

*<dstStoragePoolId>* = ID of the storage pool to use to create the checkpoint on the destination

*<dstStoragePool>* = name of the storage pool to use to create the checkpoint on the destination

*<name>*= name of the Data Mover interconnect to use for this copy session

*<interConnectId>* = ID of the Data Mover interconnect to use for this copy session

Example:

To copy checkpoint ufs1\_ckpt1 to VNX system C, type:

```
$ nas_copy -name rep1 -source -ckpt ufs1_ckpt1 -destination -pool id=3
-interconnect id=20006
```

After the copy, file system ufs1 and checkpoint ufs1\_ckpt1 are created on VNX system C.

5. Create a file system replication session for file system ufs1 from VNX system A to VNX system C by using this command syntax:

```
$ nas_replicate -name -create <name> -source -fs <fsName> -destination
-fs <existing_dstFsname> -interconnect id=<interConnectId>
```

where:

*<name>* = name of the replication session

*<fsName>* = name of the existing source file system

*<existing\_dstFsName>* = name of the existing file system on the destination

*<interConnectId>* = ID of the Data Mover interconnect to use for this replication session

Example:

To create a file system replication session, type:

```
$ nas_replicate -name rep2 -source -fs ufs1 -destination -fs ufs2 -interconnect
id=20007
```

The common base (ufs1\_ckpt1) is automatically identified and the transfer is started from a differential copy.

## Change Control Station IP addresses

Use this procedure to change the IP address on the Control Station of the swing VNX system that is used to transport the baseline copy of a source file system to the new destination location:

1. At the CLI, log in to the Control Station as the root user.

- Configure an interface for the Control Station's IP address by using the command syntax:

```
# /usr/sbin/netconfig -d <device_name>
```

where:

<device\_name> = name of the ethernet interface used by the Control Station for the public LAN

Example:

To configure interface eth3, type:

```
# /usr/sbin/netconfig -d eth3
```

- On the Configure TCP/IP screen, type the configuration information for the interface (IP address, Netmask IP, Default gateway IP, and Primary name server IP.)

When done, press **Enter**.

---

**Note:** Do not check the Use dynamic IP configuration (BOOTP/DHCP) box.

---

- Reboot the Control Station by typing:

```
# reboot
```

- At the CLI, log in to the Control Station as the root user.

- Verify the settings for the interface by using this command syntax:

```
#/sbin/ifconfig <device_name>
```

where:

<device\_name> = name of the ethernet interface used by the Control Station for the public LAN

Example:

To verify settings for interface eth3, type:

```
# /sbin/ifconfig eth3
```

Output:

```
eth3      Link encap:Ethernet  HWaddr 00:15:17:7B:89:F6
          inet addr:172.24.252.49  Bcast:172.24.252.255
Mask:255.255.255.0
          inet6 addr: 2620:0:170:84f:215:17ff:fe7b:89f6/64 Scope:Global
          inet6 addr: fe80::215:17ff:fe7b:89f6/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2686 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:222262 (217.0 KiB)  TX bytes:4884 (4.7 KiB)
          Base address:0x2020 Memory:81a60000-81a80000
```

The output shows that the IP address was changed to 172.24.252.49.

- Optionally, change the hostname, by using this command syntax:

```
# Hostname <new_host_name>
```

where:

<new\_host\_name> = name of the new host

Example:

To change the hostname to cs49, type:

```
# Hostname cs49
```

8. Optionally, change the domain name by using this command syntax:

```
# Domainname <new_domain_name>
```

where:

<new\_domain\_name> = name of the new domain

Example:

To change the domain name to corp2, type:

```
# Domainname corp2
```

9. Optionally, change the host name by editing the HOSTNAME field in the network file by typing:

```
# vi /etc/sysonfig/network
```

10. Optionally, change the host name by editing the hosts file by typing:

```
# vi /etc/hosts
```

11. Optionally, change the host name by editing the hostname file by typing:

```
# vi /proc/sys/kernel/hostname
```

12. Restart the network for the host name changes to take effect:

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

13. Start the NAS services if it is not already running by typing:

```
# /sbin/service nas start
```

14. Optionally, you can change the IP addresses for SPA and SPB by using the clariion\_mgmt command:

```
# /nasmcd/sbin/clariion_mgmt -start -spa_ip <SPA public IP address> -spb_ip  
<SPB public IP address> -use_proxy_arp
```

This process will restart both the SPA and the SPB components. You may need to wait approximately 15–20 minutes for this process to finish.

15. Verify that the IP addresses for SPA and SPB are correct by typing:

```
# /nasmcd/sbin/clariion_mgmt -info
```

If the SP's public IP addresses were typed incorrectly, update the addresses by using the command syntax:

```
# /nasmcd/sbin/clariion_mgmt -modify -spa_ip <SPA public IP address> -spb_ip
<SPB public IP address>
```

After an internal IP address has been changed to a public IP address, it cannot be reverted by using the previous command. An alternative method will have to be taken to revert (for example, connect to one of the SPs by using Ethernet or dial-in, change the network information, restart the SP, and then repeat the process for the other SP).

16. Delete the cable check file:

```
# rm /etc/rc.d/rc3.d/S95_cable_check -f
```

If you do not see to the cable check file, repeat steps 14 and 15 and see if you can access Unisphere.

17. Reboot the Control Station by typing:

```
# reboot
```

18. When the system comes back up, open a Web Browser and access Unisphere by using the Control Station IP address:

```
http://<Control Station IP Address>
```

Example

```
http:// 100.22.200.57
```

**Note:** You may see Hardware Status warnings once you log in. This is a due to the fact that the standby power supply (SPS) is still charging. The messages should disappear in 30–60 seconds.

## Perform an initial copy by using the tape transport method

Use this procedure to create a replication session and back up the primary file system from the source site by using the tape transport method.

[Performing an initial copy by using disk or tape on page 54](#) describes how the tape transport method works and provides prerequisite information.

This method is not supported for VDM replications.

**Note:** This special backup is used only for transporting replication data.

To perform an initial copy by using the tape transport method:

1. From the source side, create a file system replication session with the tape copy option by typing:

```
$ nas_replicate -create <name> -source -fs <fsName>
-destination -fs <dstFsName> -interconnect <name>|id=<InterConnectId>
-max_time_out_of_sync <maxTimeOutOfSync> -overwrite_destination -tape_copy
```

where:

`<name>` = name of replication session

`<fsName>` = name of source file system

`<dstFsName>` = name of the destination file system

`<name>` = name of the interconnect to use for this session

`<InterConnectId>` = ID of the interconnect to use for this session

`<maxTimeOutofSync>` = the `max_time_out_of_sync` in minutes

2. Use your normal backup procedure to back up the source file system as follows:
  - a. If the DMA supports NDMP environment variables, specify `TS=Y` as an environment variable. Otherwise, specify the mount point for the source file system in the backup definition with the prefix `/.ts`

For example, if the primary file system is mounted on mountpoint `/replication_pfs`, and the DMA supports NDMP environment variables, the backup definition appears as:

- Environment variable: `TS=Y`
- Backup file system: `/replication_pfs`

- b. If the DMA does not support NDMP environment variables, the backup definition appears as:

Backup file system: `/.ts/replication_pfs`

---

**Note:** `.ts` is the default value for the `server_param NDMP tapeSilveringStr`. You can change this value at any time.

---

*Configuring NDMP Backups on VNX* provides information about environment variables. For information about how to set variables, read your backup software vendor's documentation.

---

**Note:** The source file system and checkpoints must be mounted on the NDMP Data Mover.

---

3. (Optional) For one-to-many configurations, use the environment variable `REP_NAME` to specify the name of the replication session for tape transport. For example, the Backup Selections for NetBackup appear as follows:

```
SET TS=y
SET REP_NAME=<replication_name>
/replication_pfs
```

4. Transport the backup tapes to the destination site.

Move the backup catalog for the backup (of previous step) from the source site to the destination site. Catalog move is dependent on the specific DMA used. The specific DMA vendor documentation provides details on how to move the catalog.



- At the destination site, use your normal restore procedure to restore the backup.

From the backup catalog, select the file-level entry that corresponds to the backed-up checkpoint and perform a restore to the destination file system. Select the destination file system name in the restore definition by specifying:

```
/<replication_destination>
```

where:

```
<replication_destination> = mountpoint of the destination file system
```

---

**Note:** The file system must be restored to the NDMP Data Mover.

---



---

**Note:** After the restore, ensure the destination file system is mounted read-only (RO).

---

- Start the replication session by typing:

```
$ nas_replicate -start {<name>|id=<session_id>} -overwrite_destination
```

where:

```
<name> =name of the replication session
```

```
<session_id> =ID of the replication session
```

---

## Change the NDMP server\_param value

The server\_param NDMP tapeSilveringStr has the default value of ts. To change this parameter, specify a new value up to eight, alpha-numeric characters, and then restart the Data Mover for the value to take effect. For example, to change the parameter value from ts to abc, type:

```
server_param server_2 -f NDMP -modify tapeSilveringStr -value "abc"
```

After the parameter is changed, you can update the backup definition to the new value (abc). For example, the new backup definition would appear as:

```
/.abc/replication_pfs
```

---

## Recover from a corrupted file system using nas\_fsck

This section describes how to use the nas\_fsck command to check and repair a file system when replication is threatened. You run the command to check the source file system while replication is running. If nas\_fsck detects inconsistencies in the primary file system, the changes that occur as a result of using nas\_fsck are replicated to the destination file system like any other file system modifications.

---

**Note:** You cannot use `nas_fsck` if the destination file system is corrupted as a result of an improper `nas_copy` operation.

---

If corruption occurred on the source file system, perform these steps on the source side. Alternatively, if the destination file system is unstable or possibly so corrupted it might fail before the `nas_fsck` changes are replicated, perform these steps on the destination side:

1. Stop the replication by using this command syntax:

```
$ nas_replicate -stop {<name>|id=<sessionId>}
```

where:

`<name>` = name of the replication session

`<sessionId>` = ID of the replication session

2. Run `nas_fsck` using this command syntax:

```
$ nas_fsck -start <fs_name>|id=<fs_id>
```

where:

`<fs_name>` = source or destination file system name

`<fs_id>` = source or destination file system ID

---

**Note:** Running `nas_fsck` repairs corruption on the source file system, bringing it into a consistent, but not original, state. While `nas_fsck` runs, the file system is not mounted to avoid system instability. When the command is complete and inconsistencies addressed, the file system is brought back online.

---

3. When `nas_fsck` completes, start the replication again by using this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

---

## Manage expected outages

This section describes how to handle anticipated outages on the source and destination VNX system or any network outages. When planning for an outage, for instance, to restart a secondary Data Mover or to conduct standard network maintenance involving the secondary file system, follow the steps in this section to protect replication.

### Anticipated source site outages

If the outage is expected for a short period, you may choose to do nothing. In situations in which an outage is expected for a long period and the replication service continues running,

the SavVol might become full that eventually leads to an inactive session. If this scenario is likely, before the outage, follow this procedure:

1. From the source site, perform a replication switchover by using this command syntax:

```
$ nas_replicate -switchover {<name>|id=<session_id>}
```

where:

<name> = name of the replication session

<session\_id> = ID of the replication session

2. After the outage is over, start the replication session in the reverse direction by using this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>} -reverse
```

where:

<name> = name of the replication session

<session\_id> = ID of the replication session

3. Reverse the replication session by using this command syntax:

```
$ nas_replicate -reverse {<name>|id=<session_id>}
```

where:

<name> = name of the replication session

<session\_id> = ID of the replication session

#### Anticipated destination site outage

Begin by evaluating the outage period and whether the site can survive it. If the data queues easily in the SavVol, nothing need be done. For example, if the planned outage period is 1 day, the SavVol is 100 MB, the file system is 1 GB, and 200 MB of modifications occur daily, then survival is ensured for a half-day because the SavVol will fill in 12 hours. On the other hand, if only 100 MB in modifications occur daily, a day's worth of changes are protected.

If a short period could stretch into a longer outage, perform the following:

1. From the source site, stop the replication session by using this command syntax:

```
$ nas_replicate -stop {<name>|id=<session_id>}
```

where:

<name> = name of the replication session

<session\_id> = ID of the replication session

2. After the outage is ceased and the destination side is available, start the replication session by using this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

---

**Note:** In this case, the changes will still be accumulating on the source, but they will not attempt to transfer.

---

## Manage unexpected outages

This section describes how to manage unplanned VNX system source outages such as power interruptions or reboots and unanticipated VNX system destination platform or network outages.

### Unanticipated source site outages

First, decide whether to activate the Data Recovery (DR) site. If you do, consider the following:

- ◆ If the source resides on a VNX for block system that lost power, and replication is inactive, start the replication session.
- ◆ If the source resides on a Symmetrix system that lost power—where replication should still be active—no remediation is necessary.
- ◆ If only the source is down and you do not want to activate the DR site, no remediation is necessary.

If it is a short outage, you may choose to do nothing. If the administrator anticipates a long outage, run failover from the destination side as directed by the following procedure:

1. Run a failover replication to activate the DR site by using this command syntax:

```
$ nas_replicate -failover {<name>|id=<session_id>}
```

where:

`<name>` = source file system name

`<session_id>` = destination file system name

2. After the outage is over, start the replication session in the reverse direction by using this command syntax:

```
$ nas_replicate -start {<name>|id=<session_id>} -reverse
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

3. Return to your original configuration with a reverse replication by using this command syntax:

```
$ nas_replicate -reverse {<name>|id=<session_id>}
```

where:

`<name>` = name of the replication session

`<session_id>` = ID of the replication session

#### **Unanticipated destination site or network outages**

Most destination site outages require no action. For instance, if the VNX system restarts or goes offline—and the system does not fall out of sync—no remediation is necessary. Consider the following:

- ♦ If `nas_replicate -list` shows that replication is still active after the unplanned destination outage is finished, nothing needs to be done.
- ♦ If `nas_replicate -list` shows that replication is inactive and out of sync after the unplanned destination outage is finished, start the replication.

When an outage strikes your secondary Data Mover or the network connection to your secondary file system is offline without notice, follow the steps described in [Manage expected outages on page 186](#) for an anticipated destination site outage.



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

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

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

Topics included in this chapter are:

- ◆ [EMC E-Lab Interoperability Navigator on page 192](#)
- ◆ [Error messages on page 192](#)
- ◆ [Log files on page 192](#)
- ◆ [Return codes for nas\\_copy on page 193](#)
- ◆ [Interconnect validation failure on page 195](#)
- ◆ [Replication fails with quota exceeded error on page 195](#)
- ◆ [Two active CIFS servers after failover or switchover on page 195](#)
- ◆ [Network performance troubleshooting on page 196](#)
- ◆ [EMC Training and Professional Services on page 197](#)

---

## EMC E-Lab Interoperability Navigator

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

---

## Error messages

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

To view message details, use any of these methods:

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

---

## Log files

The following log files are available to help troubleshoot VNX Replicator. Additionally, you can use the `nas_replicate -info` and `nas_task -info` commands to get more information about the replication session:

- ◆ `server_log`
- ◆ `sys_log`
- ◆ `cli.log`



- ◆ nas\_log.ai.mgmt

---

### server\_log messages

The server\_log contains messages generated by a Data Mover that performs replication at the source site. To view server\_log messages, type: server\_log server\_<x>, where <x> is the name of the Data Mover. For example, server\_log server\_2.

---

### sys\_log messages

Messages will be sent to the /nas/log/sys\_log if the network goes down, a task goes into the inactive state, or the SLA cannot be met. For example, if the max\_time\_out\_of\_sync value is set to 10 minutes and the data transfer does not complete within this value, a message will be sent to the sys\_log.

Use the nas\_logviewer command to display the event log and other logs created by nas\_eventlog. For example, to view the contents of the sys\_log, type:

```
$ nas_logviewer /nas/log/sys_log|more
```

---

### cli.log messages

The cli.log file contains start and end time log entries for a particular CLI replication operation. To view the cli.log messages navigate to: /nas/log/webui/cli.log.

---

### nas\_log.ai.mgmt

The mgmt log contains runtime information associated with Command Service Daemon (mgmt) activities. To view nas\_log.ai.mgmt messages, go to nas/log/nas\_log.ai.mgmt.

---

## Return codes for nas\_copy

Table 9 on page 193 lists the possible message IDs that can be returned when running the nas\_copy command. This list can be helpful when error checking scripts that use nas\_copy. To obtain detailed information about a particular error, use the nas\_message -info <error\_id> command.

**Table 9. Return codes for nas\_copy**

Message ID	Description
OK	Command completed successfully.
13160415236	The object to be accessed is currently in use by replication or by a task. It could mean that the secondary dpinitis is not started.
13160415237	Some of the system components were not found.

**Table 9. Return codes for nas\_copy (continued)**

Message ID	Description
13160415238	This command cannot be executed because it is not supported.
13160415247	The replication name is already in use.
13160415268	An error occurred while accessing the replication database.
13160415302	The file system is not mounted.
13160415303	The specified Data Mover interconnect or its peer could not be found.
13160415304	The file system is not found/mounted, or the VDM is unloaded/not found.
13160415305	The file system size does not match.
13160415307	The object is not in the correct state for replication.
13160415315	The maximum number of allowed full copies is in progress. The maximum number is 16.
13160415324	The destination object is not empty and a common base does not exist, or the destination object is different from the common base.
13160415635	The destination is in use by another replication process.
13160415636	The maximum replication session count has been reached. The maximum number is 1024 per Data Mover.
13160415654	The destination file system is not a UNIX file system.
13160415655	You attempted to perform an FS Copy from a writeable checkpoint. The from base cannot be a writable checkpoint.
13160415657	The task has been aborted successfully.
13160415658	The task has been aborted locally.
13160415742	Destination file system or required snaps are missing.
13160415777	Refresh copy on read only file system.
13160415778	Copy check replica service return arguments.
13160415779	Failure on getting file system object from file system ID.
13160415780	Inconsistent type in processing copy abort replica request.
13160415781	Stamping secondary response error.
13160415782	No XML data in replica service response.
13160415783	Replica service response does not have destination snap information.
13160415784	Replica service response does not have file system ID information.

Table 9. Return codes for nas\_copy (continued)

Message ID	Description
13160415839	The file system's FLR type on the Control Station does not match the file system's FLR type on the source Data Mover.
13160415840	The FLR-C—enabled destination file system contains protected files.
13160415841	The FLR type of the source file system does not match the FLR type of the destination file system.
13160415842	The FLR type of the source file system does not match the FLR type of the destination file system.
13160415843	The FLR-C—enabled destination file system or the original source file system contains protected files.

## Interconnect validation failure

If you encounter the problem where replication creation sessions hang and interconnect validation fails with the following error, even though `nas_cel -interconnect -info` reports the status as OK, use the `nas_cel -update` command to recover:

```
Error 13160415446: DpRequest_dic_cannotAuthenticate
```

This command updates the local Data Mover-to-Data Mover authentication setup. For the remote VNX system, it updates all Data Movers that were down or experiencing errors during the `-create`, and restores them to service using the configuration required for Data Mover authentication.

## Replication fails with quota exceeded error

If you encounter the problem where replication fails with the following error, it means that there are too many checkpoints on the system causing the system to exceed the space allocated for the checkpoint SavVols:

```
RepV2 Failure. error: 13421840574: Disk space:quota exceeded
```

[Change the percentage of system space allotted to SavVols on page 174](#) provides the procedure to change the percentage of system space allotted to SavVols.

The *Parameters Guide for VNX for File* provides more detailed procedures for changing the percentage of system space allotted to SavVols.

## Two active CIFS servers after failover or switchover

Two active CIFS servers with the same server name and different IP addresses may result if you perform a failover or switchover operation on the second session in a VDM cascade configuration. In an A --> B --> C cascade, session 2 is the B --> C session.

To recover:

1. Determine which server you want to remain as the source.
2. Stop the non-source server. For example, if server A is the desired source, stop the session between B->C. If server C is the desired source, stop the session between A->B.
3. Stop any file system sessions associated with the VDM on the non-source server.
4. Mount the file systems that are associated with the VDM as read-only.
5. Change the status of the VDM from loaded to mounted.
6. Start the VDM session. If server A was the desired source, start the session between B->C. If server C is the desired source, start the session between A->B.
7. Start any file system sessions associated with the VDM. If server A was the desired source, start the session between B->C. If server C is the desired source, start the session between A->B.

## Network performance troubleshooting

When you experience a performance issue during the transferring of data, check the following:

- ◆ Duplex network configuration mismatch (Full, Half, Auto, and so on) between the Data Mover and the network switch.
- ◆ Packet errors and input/output bytes using the `server_netstat -i` command on the Data Mover.
- ◆ Packet errors and input/output bytes on the network switch port.
- ◆ Transfer rate in the `nas_replicate -info` command output.

### Checking network performance

To check the network performance between Data Movers:

1. Create a file on the source file system that is at least 1 MB.
2. Refresh the replication session.
3. After the refresh has completed, execute the `nas_replicate -info` command and check the Previous Transfer Rate line.

Output:

```

ID =
94_HK190807290035_0000_94_APM00062204462_0000
Name = repl
Source Status = OK
Network Status = OK
Destination Status = OK
Last Sync Time = Thu Jan 24 08:18:42 EST 2008
Type = filesystem
Celerra Network Server = reliant
Dart Interconnect = e2
Peer Dart Interconnect = r2
Replication Role = source
Source Filesystem = ent_fs1
Source Data Mover = server_2
Source Interface = 172.24.188.141
Source Control Port = 0
Source Current Data Port = 0
Destination Filesystem = ent_fs1
Destination Data Mover = server_2
Destination Interface = 172.24.188.153
Destination Control Port = 5085
Destination Data Port = 8888
Max Out of Sync Time (minutes) = 10
Next Transfer Size (KB) = 0
Current Transfer Size (KB) = 0
Current Transfer Remain (KB) = 0
Estimated Completion Time =
Current Transfer is Full Copy = No
Current Transfer Rate (KB/s) = 9891
Current Read Rate (KB/s) = 2159
Current Write Rate (KB/s) = 2602
Previous Transfer Rate (KB/s) = 0
Previous Read Rate (KB/s) = 0
Previous Write Rate (KB/s) = 0
Average Transfer Rate (KB/s) = 9891
Average Read Rate (KB/s) = 0
Average Write Rate (KB/s) = 0

```

## EMC Training and Professional Services

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

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



## Setting up the CIFS replication environment

Setting up a CIFS replication environment means your source and destination sites have an identical starting point. For CIFS data recovery implementation, a fully functional CIFS environment has to be available at the source and destination sites. The environment includes:

- ◆ Domain controllers
- ◆ Name resolution service (for example, DNS or WINS)
- ◆ Time service synchronized between the domain controllers and clients
- ◆ Network interfaces
- ◆ User mapping
- ◆ Client access to the source and destination VNX systems

---

**Note:** You should have at least two DNS and NTP servers to avoid single point of failure.

---

[Table 10 on page 199](#) lists the environment variables used in the examples in this section.

**Table 10. Environment variables for CIFS replication**


Variable	Source	Destination
Control Station	Prod_Site (Boston)	DR_Site (Miami)
Domain	c1t1.pt1.c3lab.nsgprod.emc.com	c1t1.pt1.c3lab.nsg-prod.emc.com
Domain controller	172.24.100.183	172.24.101.222
DNS	172.24.100.183	172.24.101.222
NTP	172.24.100.183	172.24.101.222

**Table 10. Environment variables for CIFS replication (continued)**

Variable	Source	Destination
Internal Usermapper	Secondary	Primary: server_2
VDMs	Engineering	Engineering
CIFS servers	ENG_NE	ENG_NE
Network interface for client data access	Eng - 172.24.100.125	Eng - 172.24.101.38
Network interface for Replicator and Internal Usermapper	Ace0 - 172.24.106.21	S2_fsn2 - 172.24.106.145

Table 11 on page 200 lists the interfaces in a CIFS environment.

**Table 11. Interfaces in a CIFS environment**

Interface	Source site	Destination site
Interfaces for source CIFS servers	(Required) Must be identical to the interface names on the destination site.	(Required) Must be identical to the interface names on the source site.
	 If the interface names on the sites are not identical, you will not be able to load the VDM on the destination Data Mover after a failover.	
Interconnect interfaces used for replication	(Optional) You can use different interfaces to transmit the replication data.  The names of these Data Mover interconnect interfaces do not have to be identical on the source and destination sites.	(Optional) You can use separate interfaces to transmit the replication data.  The names of these interfaces do not have to be identical on the source and destination sites.
<p><b>Note:</b> For every CIFS server being replicated in a VDM, you must have that CIFS server's interfaces available and identically named on the destination site.</p>		

The tasks to configure a CIFS replication environment and to set up your file systems for replication are as follows. *Configuring and Managing CIFS on VNX* contains complete CIFS procedures:



- ◆ [Verify IP infrastructure on page 202](#)
- ◆ [Configure an interface on page 204](#)
- ◆ [Set up DNS on page 206](#)
- ◆ [Synchronize Data Mover and Control Station Time on page 207](#)
- ◆ [Configure user mapping on page 210](#)
- ◆ [Prepare file systems for replication on page 211](#)

## Verify IP infrastructure

You should verify that the IP infrastructure has been set up correctly for CIFS data recovery.

1. Verify that there is communication between the source and destination VNX systems.

[Chapter 4](#) provides details on how to set up and verify communication between VNX systems.

2. Initialize interface connectivity.

The network connectivity to support replication and user access to the data in a failover must be present on the destination site. IP connectivity is required for all network interfaces used in the replication relationship.



The interface names assigned to the CIFS servers on the source and destination sites must be exactly the same to execute a successful failover. You can also have additional interfaces such as the interfaces on each site used by replication for transmission of changes.

3. Identify the interface names on the source site so that you can use them to set up the same interface names on the destination site. Generate the following lists:

List the CIFS servers that exist on the VDM by using this command syntax:

```
$ server_cifs <movename>
```

where:

*movename* = name of the Data Mover on the source site

Output:

```

$ server_cifs server_2
server_2:
256 Cifs threads started
Security mode = NT
Max protocol = NT1
I18N mode = UNICODE
Home Directory Shares DISABLED
Usermapper auto broadcast enabled
Usermapper[0] = [128.221.252.2] state:active port:12345
(auto discovered)
Usermapper[1] = [128.221.253.2] state:active (auto discovered)
Enabled interfaces: (All interfaces are enabled)
Disabled interfaces: (No interface disabled)
Unused Interface(s):
  if=Ace0 l=172.24.106.21 b=172.24.106.255 mac=0:60:16:b:f8:11
-----
CIFS service of VDM Engineering (state=loaded)
Home Directory Shares DISABLED

DOMAIN C1T1 FQDN=clt1.pt1.c3lab.nsgprod.emc.com SITE=Default-First-
Site-Name RC=3
  SID=S-1-5-15-b23c567a-11dc6f47-d6c3b7d5-ffffffff
>DC=C1T1-DC1(172.24.100.183) ref=3 time=1 ms (Closest Site)

CIFS Server ENG-NE[C1T1] RC=2
Full computer name=eng-ne.clt1.pt1.c3lab.nsgprod.emc.com
realm=C1T1.PT1.C3LAB.NSGPROD.EMC.COM
Comment='EMC-SNAS:T5.6.39.1'
if=Eng l=172.24.100.125 b=172.24.100.255 mac=0:60:16:b:f8:11
  FQDN=eng-ne.clt1.pt1.c3lab.nsgprod.emc.com (Updated to DNS)
Password change interval: 720 minutes
Last password change: Tue Sep 30 15:43:35 2008 GMT
Password versions: 2

```

---

**Note:** The output shows that Eng is the interface name that must be identical to the interface name on the destination site.

---

- List the interface names used by the CIFS servers on the VDM by using this command syntax:

```
$ server_ifconfig <movertname> -all
```

Example:

```
$ server_ifconfig server_2 -all
```

Output:

```

server_2 :
Eng protocol=IP device=cge0
  inet=172.24.100.125
    netmask=255.255.255.broadcast=172.24.100.255
  UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:b:f8:11

Ace0 protocol=IP device=cge1
  inet=172.24.106.21 netmask=255.255.255.0 broadcast=172.24.106.255
  UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:b:f8:11

loop protocol=IP device=loop
  inet=127.0.0.1 netmask=255.0.0.0 broadcast=127.255.255.255
  UP, loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0
  netname=localhost

el31 protocol=IP device=mge1
  inet=128.221.253.4 netmask=255.255.255.0 broadcast=128.221.253.255
  UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:b:b7:1f
  netname=localhost

el30 protocol=IP device=mge0
  inet=128.221.252.4 netmask=255.255.255.0 broadcast=128.221.252.255
  UP, ethernet, mtu=1500, vlan=0, macaddr=0:60:16:b:b7:1d
  netname=localhost

```

### After you finish

In most CIFS data-recovery configurations, your network addresses (IPs) change because your data recovery site resides in a different subnet. Windows Server 2000 with dynamic DNS masks this change from your clients. Using UNC pathnames, when the VDM is loaded, the CIFS servers dynamically declare in the DNS and the records are updated as soon as they start up on the destination site in a failover. The speed at which these changes propagate through the DNS servers used by your clients depends on your environment.

If you want to retain the same IP addresses on the destination site in a failover, you may do so, but you will need to configure your interfaces on the destination site and they must be down until you are ready to activate the destination site.

*Configuring and Managing Networking on VNX and Configuring and Managing Network High Availability on VNX* describe network interface setup in detail.

## Configure an interface

You can create one interface for data access by one group and a separate interface for VNX Replicator and Internal Usermapper.

Action
<p>To configure an interface, use this command syntax:</p> <pre>\$ <b>server_ifconfig</b> &lt;movername&gt; <b>-create -Device</b> &lt;device_name&gt; <b>-name</b> &lt;if_name&gt; <b>-protocol IP</b> &lt;ipaddr&gt; &lt;ip_mask&gt; &lt;ipbroadcast&gt;</pre> <p>where:</p>

**Action**

`<movername>` = name of the physical Data Mover  
`<device_name>` = name of the device  
`<if_name>` = name of the interface  
`<ipaddr> <ip_mask> <ipbroadcast>` = IP protocol that includes IP address, subnet mask, and broadcast address

Example:

To configure a network interface to service clients on the source, type:

```
$ server_ifconfig server_2 -create -Device s2_trk1 -name Eng -protocol IP
172.24.100.125 255.255.255.0 172.24.100.255
```

To configure a network interface for the replication data, type:

```
$ server_ifconfig server_2 -create -Device ace0 -name ace0 -protocol IP 172.24.106.21
255.255.255.128 172.24.106.127
```

Next, configure an interface on the destination.

To configure a network interface to service clients on the destination, type:

```
$ server_ifconfig server_2 -create -Device s2_trk1 -name Eng -protocol IP
172.24.101.38 255.255.255.0 172.24.100.255
```

To configure a network interface for the replication data, type:

```
$ server_ifconfig server_2 -create -Device ace0 -name ace0 -protocol IP
172.24.106.145 255.255.255.128 172.24.106.127
```

**Output**

```
server_2 : done
```

**Verification**

To verify that the interface on the source site used for data transmission by VNX Replicator and Internal Usermapper can communicate with the interface on the destination site, type:

```
$ server_ping server_2 -interface ace0 172.24.106.145
```

Output:

```
server_2 : 172.24.106.145 is alive, time= 3 ms
```

**After you finish**

You should verify that connectivity exists between the Data Movers used in the replication session. [Validate Data Mover communication on page 99](#) explains how to verify communication between Data Movers. If no connectivity exists, [Chapter 5](#) provides details on how to set up communication between Data Movers.

## Set up DNS

### Before you begin

Before setting up DNS, you must have a proper high-availability DNS architecture on both sites of the environment.

### Procedure

Set up DNS resolution on the source site and then on the destination site as follows:

1. Set up DNS on the source site by using this command syntax:

```
$ server_dns <movername> -protocol [tcp|udp] <domainname> {<ip_addr>, ...}
```

where:

<movername> = name of the physical Data Mover

<domainname> = domain controller on the source site

<ip\_addr> = IP address on the source site

Example:

To set up DNS on the source VNX system, type:

```
$ server_dns server_2 -protocol udp c1t1.pt1.c3lab.nsgprod.emc.com 172.24.100.183
```

---

**Note:** Using Active Directory integrated DNS services speeds the propagation of dynamic DNS updates, but is unnecessary. Use any DNS framework compliant with Windows and VNX system.

---

### Output

```
server_2 : done
```

2. To verify that DNS is running on the source site, type:

```
$ server_dns server_2
```

Output:

```
server_2 :
DNS is running.
c1t1.pt1.c3lab.nsgprod.emc.com
proto:udp server(s):172.24.100.183
```

3. Set up DNS on the destination site by using this command syntax:

```
$ server_dns <movername> -protocol [tcp|udp] <domainname> {<ip_addr>, ...}
```

where:

<movername> = name of the physical Data Mover

<domainname> = domain controller on the destination site

<ip\_addr> = IP address on the destination site

Example:

To set up DNS on the destination VNX system, type:

```
$ server_dns server_2 -protocol udp c1t1.pt1.c3lab.nsgprod.emc.com 172.24.101.222
```

---

**Note:** You can have multiple DNS servers.

---

Output:

```
server_2 : done
```

4. To verify that DNS is running on the destination side, type:

```
$ server_dns server_2
```

Output:

```
server_2 :
DNS is running.
c1t1.pt1.c3lab.nsgprod.emc.com
proto:udp server(s):172.24.101.222
```

---

## Synchronize Data Mover and Control Station Time

The Data Movers and Control Stations in a replication relationship must have synchronized times. The maximum allowable time skew is 10 minutes.

---

### Synchronize Data Mover source site time

#### Action

To synchronize time services on the Data Mover at the source site, use this command syntax:

```
$ server_date <movername> timesvc start ntp <host>
```

where:

<movername> = name of the physical Data Mover

<host> = IP address of the source site domain controller

Example:

To synchronize time services for the source VNX system, type:

```
$ server_date server_2 timesvc start ntp 172.24.100.183
```

This example shows synchronization only for the Data Movers.

---

**Note:** Windows Server domain controllers automatically service time requests and also synchronize time between them. You synchronize the two sites to their respective domain controllers.

---

**Output**

```
server_2 : done
```

**Verification**

To verify that the Data Mover contacted the NTP server, type:

```
$ server_date server_2 timesvc stats
```

```
server_2 :
Time synchronization statistics since start:
hits= 120, misses= 1, first poll hit= 1, miss= 2
Last offset: 0 secs, -43507 usecs
Current State: Running, connected, interval=60
Time sync hosts:
    0 1 172.24.100.183
```

---

## Synchronize Data Mover destination site time

**Action**

To synchronize time services on the destination site, use this command syntax:

```
$ server_date <movername> timesvc start ntp <host>
```

where:

<movername> = name of the physical Data Mover

<host> = IP address of the destination site domain controller

Example:

To synchronize time services for the destination VNX system, type:

```
$ server_date server_2 timesvc start ntp 172.24.101.222
```

**Note:** Ensure that all time servers and clients accessing the domain controller within the domain are synchronized.

**Output**

```
server_2 : done
```

---

## Synchronize Control Station source site time

You must set the source and destination Control Stations as NTP clients to Time Services. All Data Movers and Control Stations in the replication relationship must have synchronized times. The maximum allowable time skew is 10 minutes.

To set the source Control Station as an NTP client to Time Services:



1. On the source Control Station, check if the NTP daemon is running by typing:

```
# /sbin/chkconfig ntpd --list

ntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

2. On the source and destination Control Stations, start the NTP daemon by typing:

```
# /sbin/chkconfig --level 345 ntpd on

# /sbin/chkconfig ntpd --list

ntpd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

3. On the source Control Station, edit the `/etc/ntp.conf` file to add the network NTP server and comment out the other two lines, as shown below:

```
# server 192.168.10.21 local clock

# fudge 192.168.10.21 stratum 10 [Commented out so as not to
point to itself as the primary NTP Service]
server 192.1.4.218 minpoll 8 ##NTP Server
```

4. On the source Control Station, add the IP address of the NTP servers to the `/etc/ntp/step-tickers` file:

```
192.1.4.218
```

5. On the source Control Station, restart the NTP daemon by typing:

a.

```
# /sbin/service ntpd restart
```

Output:

```
Shutting down ntpd: [ OK ]
Synchronizing with time server: [ OK ]
Starting ntpd:
```

b.

```
# ps -ef |grep ntp
```

Output:

```
ntp      15364      1  0 14:00 ?        00:00:00 ntpd -U ntp
root    16098   9015  0 14:01 pts/1    00:00:00 grep ntp
```

c.

```
# /sbin/service ntpd status
```

Output:

```
ntpd (pid 28905) is running...
```

6. On the source Control Station, manually set the date and time by typing:

```
# date -s "4/09/03 16:15:15" # /usr/sbin/setclock
```

```
date à Wed Apr 9 16:15:30 MDT 2003
```

---

## Synchronize Control Station destination site time

On the destination Control Station, repeat steps 1 through 6 in [Synchronize Control Station source site time on page 208](#) to set the destination Control Station as an NTP client to Time Services.

---

## Configure user mapping

User mapping allows you to map a Windows user or group to a corresponding UNIX user or group. After a user is mapped through a Data Mover or a VDM, that mapping information is remembered by the Data Mover and is not requested again.

It is not required that you use EMC Internal Usermapper. However, if not used, one of the other ways of using VNX to resolve Windows SID to UNIX UID/GID mappings is required, as explained in *Configuring User Mapping on VNX*.

Password and group files are not transferred in the VDM. Therefore, you must update files on the destination site when you update files on the source site. If your CIFS environment uses local password and group files, you must update the destination site whenever making any changes to these files on the source site.

To configure user mapping:

1. Verify user mapping service on the destination site.
2. Enable secondary user mapping service on the source site.
3. Verify user mapping service on the source site.

---

**Note:** You only need to verify the user mapping service on the destination and source sites if you are using EMC Internal Usermapper to do the mapping. If so, usermapping should be running as primary on the destination site and secondary on the source site. If there are multiple VNX systems, only one usermapping should be running as the primary.

---

Configure user mapping by doing the following:

1. To verify the user mapping service on the destination site (Miami), our core data center, and primary usermapper site, type:

```
$ server_usermapper server_2
```

---

**Note:** This task is required only if you are using EMC Internal Usermapper to do the mapping.

---

Output:

```
server_2 : Usrmapper service: Enabled  
Service Class: Primary
```

2. Enable the secondary user mapping service on the source site (Boston) by typing:

```
$ server_usermapper server_2 -enable primary=172.24.106.145
```

Output:

```
server_2 : done
```

3. To verify the user mapping service on the source site (Boston), type:

```
$ server_usermapper server_2
```

---

**Note:** This task is required only if you are using EMC Internal Usermapper to do the mapping.

---

Output:

```
server_2 : Usrmapper service: Enabled  
Service Class: Secondary  
Primary = 172.24.106.145 (c)
```

---

## Prepare file systems for replication

The tasks involved in preparing file systems for replication are:

1. Create a VDM
2. Create source file systems
3. Mount source file systems on the VDM
4. Create CIFS servers in the VDM
5. Join CIFS servers in the VDM to the domain
6. Create shares for the CIFS servers in the VDM

Use the procedures in the following technical modules for the specific steps:

- ♦ *Managing Volumes and File Systems with VNX Automatic Volume Management* describes AVM and creating volumes and file systems in detail.
- ♦ *Configuring Virtual Data Movers for VNX* details VDMs in depth.



### A

#### ***active replication session***

Replication session that is transferring the delta from source to destination site.

#### ***automatic file system extension***

Configurable file system feature that automatically extends a file system created or extended with AVM when the high water mark (HWM) is reached.

See also *high water mark*.

#### ***Automatic Volume Management (AVM)***

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

See also thin provisioning.

### B

#### ***bandwidth***

Maximum amount of data that can be transmitted through a data channel per unit of time. Usually expressed in megabytes per second.

#### ***bandwidth schedule***

List of time periods (days and hours) and bandwidth values (in KB/s) that control the amount of bandwidth available for a given Data Mover interconnect.

### C

#### ***checkpoint***

Point-in-time, logical image of a PFS. A checkpoint is a file system and is also referred to as a checkpoint file system or an EMC SnapSure™ file system.

See also *Production File System*.

#### ***CIFS***

See Common Internet File System.

**CIFS server**

Logical server that uses the CIFS protocol to transfer files. A Data Mover can host many instances of a CIFS server. Each instance is referred to as a CIFS server.

**CIFS service**

CIFS server process that is running on the Data Mover and presents shares on a network as well as on Microsoft Windows-based computers.

**command line interface (CLI)**

Interface for typing commands through the Control Station to perform tasks that include the management and configuration of the database and Data Movers and the monitoring of statistics for VNX for file cabinet components.

**common base**

Internal checkpoint that is common on both replication source and destination site and is used as a base in the next differential data transfer.

**Common Internet File System (CIFS)**

File-sharing protocol based on the Microsoft Server Message Block (SMB). It allows users to share file systems over the Internet and intranets.

**configured replication session**

Replication session that is idle, active, or inactive. A stopped replication session is not a configured session.

**Control Station**

Hardware and software component of VNX for file that manages the system and provides the user interface to all VNX for file components.

**Cyclic Redundancy Check (CRC)**

Method used to perform additional error checking on the data sent over a Data Mover interconnect to ensure data integrity and consistency. A CRC is a type of hash function used to produce a checksum, a small fixed number of bits, against a block of data transferred to a remote site. The checksum is used to detect errors after transmission. A CRC is computed and appended before transmission and verified afterwards to confirm that no changes occurred on transit.

**D****data access in real time (DART)**

On VNX for file, the operating system software that runs on the Data Mover. It is a realtime, multithreaded operating system optimized for file access, while providing service for standard protocols.

**Data Mover**

In VNX for file, a cabinet component that is running its own operating system that retrieves data from a storage device and makes it available to a network client. This is also referred to as a blade.

**deduplication**

Process used to compress redundant data, which allows space to be saved on a file system. When multiple files have identical data, the file system stores only one copy of the data and

shares that data between the multiple files. Different instances of the file can have different names, security attributes, and timestamps. None of the metadata is affected by deduplication.

**delta**

For Replication (V2), block changes to the source object, as calculated by comparing the newest, currently marked internal checkpoint (point-in-time snapshot) against the previously replicated internal checkpoint. Replication transfers these changes and applies them immediately to the destination object, then refreshes the latest destination internal checkpoint.

**destination VNX for file**

Term for the remote (secondary) VNX for file in an SRDF or MirrorView/S configuration. The destination VNX for file is typically the standby side of a disaster recovery configuration. Symmetrix configurations often call the destination VNX for file: the target VNX for file.

**differential copy**

The difference between the common base and the object that needs to be copied. Only the delta between the common base and the source object is transferred to the destination.

**domain**

Logical grouping of Microsoft Windows Servers and other computers that share common security and user account information. All resources such as computers and users are domain members and have an account in the domain that uniquely identifies them. The domain administrator creates one user account for each user in the domain, and the users log in to the domain once. Users do not log in to each individual server.

**Domain Name System (DNS)**

Name resolution software that allows users to locate computers on a UNIX network or TCP/IP network by domain name. The DNS server maintains a database of domain names, hostnames, and their corresponding IP addresses, and services provided by the application servers.

See also *ntxmap*.

**E****event**

System-generated message caused by a command, an error, or other condition that may require action by an administrator. Events are typically written to an event log, and may trigger an event notification.

**event log**

File of system-generated messages based on meeting a condition of a particular severity level. There are separate event logs for the Control Station and for each physical and virtual Data Mover in the system.

**F****failover**

Process of immediately routing data to an alternate data path or device to avoid interrupting services in the event of a failure. The impact to service is dependent on the application's ability to handle the change gracefully.

**file server**

Computer system that is optimized to serve files to clients. A file server does not run general purpose applications. VNX for file refers to the complete system, which includes several Data Movers and other components. The blocks of data that make up the files served to clients are stored on the system.

**file storage object**

File created on an UxFS file system that provides the storage space for an iSCSI LUN.

**file system**

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

**file system name**

Unique identifier for a file system on VNX for file. There can be only one file system with a particular name across all the Data Movers on the system.

See also *share name*.

**file-level retention (FLR)**

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

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

**G****graphical user interface (GUI)**

Software that uses graphical objects such as pull-down menus and operations such as drag-and-drop to allow the user to enter commands and execute functions.

**H****Hypertext Transfer Protocol (HTTP)**

Communications protocol used to connect to servers on the World Wide Web.

**Hypertext Transfer Protocol Secured (HTTPS)**

HTTP over SSL. All network traffic between the client and server system is encrypted. In addition, there is the option to verify server and client identities. Server identities are typically verified and client identities are not.

**I****idle replication session**

Replication session that is configured but is not transferring data from the source site to the destination site.

**inactive replication session**

Replication session that experiences a fatal error. User must stop or delete the replication session to continue.

**initial copy**

Full copy of the source data from the source site to the destination site.



**initialization**

The process of creating the original baseline copy of a source object for the purpose of establishing a replication. The baseline copy is then either transported over the network (the default behavior) or physically by tape or disk from the source site to the destination site.

See also *silvering*.

**interconnect**

Communication path between a given Data Mover pair located on the same VNX for file cabinet or different cabinets.

**interface (network)**

Named logical element mapped to a physical network connection, or port, on a Data Mover. Each interface assigns an IP address to the port.

**internal checkpoint**

Read-only, logical point-in-time image of a source object created during replication on the source and destination sides. Internal checkpoints cannot be managed by users. Replicator (V2) internal checkpoints are identified by the prefix `root_rep_ckpt`.

**Internet Protocol (IP)**

Network layer protocol that is part of the Open Systems Interconnection (OSI) reference model. IP provides logical addressing and service for end-to-end delivery.

**Internet Protocol address (IP address)**

Address uniquely identifying a device on any TCP/IP network. Each address consists of four octets (32 bits), represented as decimal numbers separated by periods. An address is made up of a network number, an optional subnetwork number, and a host number.

**Internet SCSI (iSCSI)**

Protocol for sending SCSI packets over TCP/IP networks.

**Internet Storage Name Service (iSNS)**

Discovery and naming protocol designed to facilitate the automated discovery, management, and configuration of iSCSI and Fibre Channel Protocol (FCP) devices on a TCP/IP network.

**iSCSI host**

Computer hosting an iSCSI initiator.

**iSCSI initiator**

iSCSI endpoint, identified by a unique iSCSI name, which begins an iSCSI session by issuing a command to the other endpoint (the target).

**iSCSI qualified name (IQN)**

Naming standard supported by the iSCSI protocol. IQN names are globally unique and in the form of `iqn`, followed by a date and a reversed domain name.

**iSCSI target**

iSCSI endpoint, identified by a unique iSCSI name, which executes commands issued by the iSCSI initiator.

**L*****local replication***

Replication of a file system on a single VNX for file with the source file system on one Data Mover and the destination file system on another Data Mover.

***loopback replication***

Replication of a file system with the source and destination file systems residing on the same Data Mover.

***LUN mask***

Set of access permissions that identify which iSCSI initiator can access specific LUNs on a target.

**N*****NDMP client***

Application that controls the NDMP session. The NDMP client runs an NDMP-compliant backup application, such as EMC NetWorker®.

***NDMP host***

Host system (Data Mover) that executes the NDMP server application. Data is backed up from the NDMP host to either a local tape drive or to a backup device on a remote NDMP host.

***NDMP server***

NDMP process that runs on an NDMP host, which is a Data Mover in VNX for file environment.

***Network Data Management Protocol (NDMP)***

Open standard network protocol designed for enterprise-wide backup and recovery of heterogeneous network-attached storage.

***network file system (NFS)***

Network file system (NFS) is a network file system protocol that allows a user on a client computer to access files over a network as easily as if the network devices were attached to its local disks.

***network-attached storage (NAS)***

Specialized file server that connects to the network. A NAS device, such as VNX for file, contains a specialized operating system and a file system, and processes only I/O requests by supporting popular file sharing protocols such as NFS and CIFS.

**P*****Production File System (PFS)***

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

**R*****recovery point objective (RPO)***

Describes the acceptable amount of data loss, measured in units of time, for example, 12 minutes, 2 hours, or 1 day. This represents a target that is derived from conditions specified in an SLA,

RTO, and relevant analyses. The RPO in conjunction with the recovery time objective (RTO) is the basis on which data protection strategy is developed.

***recovery time objective (RTO)***

How long a business process can be down before consequences are unacceptable. This represents a target that is derived from conditions specified in an SLA and business impact analysis.

***remote replication***

Replication of a file system from one VNX for file to another. The source file system resides on a different system from the destination file system.

***replication***

Service that produces a read-only, point-in-time copy of a source file system. The service periodically updates the copy, making it consistent with the source file system.

***replication reversal***

Process of reversing the direction of replication. The source file system becomes read-only and the destination file system becomes read/write.

**S**

***service level agreement (SLA)***

Contract or agreement that formally defines the level of expected availability, serviceability, performance, operation, or other attributes of service and even penalties in the case of violation of the SLA. An SLA may include acceptable downtime or disaster recovery time. In theory, an SLA is a formal agreement. However, in practice the agreement is often informal in which case it may be called a service level expectation (SLE).

***share***

File system, directory, or service that has been made available to CIFS users on the network. Also, the process of making a file system, directory, or service available to CIFS users on the network.

***share name***

Name given to a file system, or resource on a file system available from a particular CIFS server to CIFS users. There may be multiple shares with the same name, shared from different CIFS servers.

***silvering***

See initialization.

***snapshot***

Generic term for a point-in-time copy of data.

***source object***

The production data being copied. Also known as the Production File System, production LUN, and primary file system. Replicator source objects are file systems, iSCSI LUNS, and Virtual Data Movers (VDM).

**source VNX for file**

Term for the local (primary) VNX for File. A source VNX for file is typically the “production” side of a disaster recovery SRDF/S or MirrorView/S configuration.

**storage pool**

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

See also *Automatic volume management (AVM)*

**switchover**

Replication operation that synchronizes the destination object with the source, stops the replication with no data loss, and then mounts the destination object read/write and the source object read-only.

**T****thin provisioning**

Configurable VNX for File feature that lets you allocate storage based on long-term projections, while you dedicate only the file system resources that you currently need. NFS or CIFS clients and applications see the virtual maximum size of the file system of which only a portion is physically allocated.

See also *Automatic Volume Management*.

**throttle schedule**

See bandwidth schedule.

**time out of sync**

Maximum amount of time in minutes (1–1440), or hours that the source object and the destination object in a replication session can be out of synchronization before an update is performed.

**V****Virtual Data Mover (VDM)**

VNX for file software feature that enables users to administratively separate CIFS servers, replicate CIFS environments, and move CIFS servers from one Data Mover to another.

**W****writeable snapshot**

Read-write, point-in-time, logical image of a PFS created from a read-only (baseline) snapshot.

See also *Production File System*.

**C**

## cautions

- Unicode to ASCII replication 17

## commands

- nas\_cel, resuming interconnect 169
- nas\_replicate, deleting 133, 134
- nas\_replicate, refreshing 132
- nas\_replicate, reversing 140
- nas\_replicate, starting 136, 137, 138, 139
- nas\_replicate, starting in the reverse direction 138
- nas\_replicate, stopping 135
- server\_sysstat, monitor memory usage 60

**D**

- deleting replication 133, 134

**E**

- EMC E-Lab Navigator 192
- error messages 192

**F**

- file system copy 22
  - replication 22
- file system replication 22

**I**

## interconnect

- configuration elements 37
- resuming data transmission 169
- using name service interface names 58
- validating 169

## interface

- listing 37
- IP aliasing 60

**L**

- local replication 12, 21
- log files 192
- loopback replication 21

**M**

- manual refresh 43
- max\_time\_out\_of\_sync
  - using 132
- messages, error 192

**N**

- nas\_copy 22, 42
- nas\_fs, monitoring file systems 172
- nas\_replicate
  - monitoring replication sessions 172

**O**

- one-to-many replication configuration 27
- outages
  - managing 188

**R**

- refresh policy 43
- remote replication 12, 21
  - system requirements 12
- replication 22, 23, 27, 45, 132, 133, 134, 140
  - deleting 133, 134

- replication (*continued*)
  - file system 22
  - one-to-many 27
  - refreshing 132
  - reversing 140
  - starting a replication session 45
  - VDM 23
- replication types 20, 21
  - local replication 21
  - loopback replication 21
  - remote replication 21
- replications
  - replication types 20
  - source objects 21
- restrictions
  - system 13
- reversing replication
  - maintenance procedure 140

## S

- server\_df, monitoring file systems 172
- server\_sysstat, monitoring replication 172
- source objects 24
- start

- start (*continued*)
  - stopped replication 45
- system
  - requirements 12
  - restrictions 13
- system requirements 12

## T

- troubleshooting 191

## U

- update policy
  - manual 44
  - max\_time\_out\_of\_sync 44
- updating the destination 43

## V

- VDM replication 23
- VNX Replicator 12, 13, 59, 192
  - log files 192
  - planning 59
  - restrictions 13